



# LUND UNIVERSITY

## Dynamic Model Predictive Control

Mårtensson, Karl; Wernrud, Andreas

2008

[Link to publication](#)

*Citation for published version (APA):*

Mårtensson, K., & Wernrud, A. (2008). *Dynamic Model Predictive Control*. Paper presented at 17th IFAC World Congress, 2008, Seoul, Korea, Democratic People's Republic of.

*Total number of authors:*

2

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Dynamic Model Predictive Control

Karl Mårtensson<sup>\*,\*</sup> Andreas Wernrud<sup>\*,\*</sup>

*\* Department of Automatic Control, Faculty of Engineering, Lund University, Box 118, SE 221 00 Lund, Sweden.  
E-mail: {karl, andreas}@control.lth.se*

---

**Abstract:** In this paper an alternative approach to model predictive control is presented. In traditional MPC a finite horizon open loop optimal control problem is solved in each sampling instance. When uncertainties such as computational delays are present, one can encounter problems. We propose to parametrize the control sequence in each sampling instant in terms of a linear feedback controller, i.e. in each sample a dynamic feedback compensator is computed. Thus, if computational delays are present the control system runs in closed loop, decreasing the need for ad hoc solutions used in traditional MPC.

---

## 1. INTRODUCTION

Model predictive control (MPC), see, e.g. Maciejowski (1991); Garcia, et al. (1989), has been used by the industry for several years, e.g. in the chemical industry and other industries with processes with slow dynamics. The main reason for the success of MPC is the ability to control constrained multivariable systems. The MPC controller often relies on an online solution of a finite horizon optimal control problem, in *each* sample. Usually, the optimal control problem is reformulated as finite dimensional convex optimization problem. For example, a linear system with linear constraints on states and control variables and a quadratic step cost can be formulated as quadratic optimization problem. There is a rich class of convex optimization problems which are guaranteed to be solvable in polynomial time. Even with increasing computer power, on systems with fast dynamics this might not be fast enough for the MPC scheme. Since if the system must be sampled too fast the MPC controller will not be able to finish its required computations in time. This is one reason why model predictive control is most successful on slow processes.

Moreover, in practice the optimization solver can have quite unpredictable execution times, the computations can exceed the time of one sample or even a few samples. This obviously leads to reduced performance of the system and might even lead to instability. One approach to handle such problems has been studied in Henriksson, D. (2006), see also the references therein.

To compensate for such computational delays, a new approach is presented in this paper. Instead of calculating an optimal sequence of control inputs, an optimal dynamic controller is computed in each sampling instance. With the use of the Youla-parametrization, Youla, et al. (1976a,b), the MPC problem in its original form can be reformulated to depend on the parameters of the controller in such a way that the optimization problem can be solved with the same complexity as the original problem. Without computational delays, the resulting dynamic controller can

be shown to be equivalent with the controller obtained from the original MPC formulation. Also, with this setup, when computational time delays occur, there is now a feedback controller which controls the process. This will improve performance since the system operates in closed loop at all times. This is contrary to traditional MPC, which will operate in open loop.

Work that relate to ours can be found in Lofberg (2003) where the author studies robust MPC. This approach was later extended in Goulart, et al. (2006). These two references consider problems that are similar to the ones considered in this paper, but our approach is different from theirs.

The outline of the paper is as follows: In section 2 a short introduction to traditional MPC is given. The main contribution of the paper is presented in section 3. The structure of the controller is described and the MPC formulation is revised to fit the new structure. Difficulties that arises when computational delays are introduced, are treated. Two case studies are presented in section 4. In the first example a double integrator is studied. In the second example the method of Dynamic MPC is applied on a DC-DC converter (for a thorough description of DC-DC converters see e.g. Wernrud (2008)). In section 5 a short summary of the required on-line computation needed in Dynamic MPC is given.

## 2. TRADITIONAL MPC

Consider the discrete, time-invariant, linear plant  $P$

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma_1 w(k) + \Gamma_2 u(k) \\z(k) &= C_1 x(k) + D_{11} w(k) + D_{12} u(k) \\y(k) &= C_2 x(k) + D_{21} w(k)\end{aligned}$$

where  $u$  is the control input and  $w$  the disturbances. The  $z$  vector are referred to as the controlled output, it contains those signals that we will include in the system performance index and those that we will put constraints on. For example states, tracking errors or control variables. The output  $y$  is the measured output that can be used for feedback. Notice that we do not allow a direct term of the control input to  $y$ . This is due to the fact that to

---

\* Authors are listed in alphabetic order.

determine  $u(k)$ , a measurement of  $y(k)$  is needed and hence  $y(k)$  cannot depend on  $u(k)$ .

In model predictive control a finite horizon optimal control problem is solved in each sample. The cost function is defined as

$$V(x(0), \mathbf{u}) = \sum_{i=0}^{N-1} \ell(z(i)) + F(x(N)) \quad (1)$$

where  $z(i)$  are the controlled outputs of  $P$  when the control sequence  $\mathbf{u} = (u(0), \dots, u(N-1))$  is applied to the system with initial state  $x(0)$ . In this paper we assume full state feedback, and hence the state  $x(0)$  is known. Since the system is time-invariant it can be assumed that the initial time always is  $k = 0$ . The functions  $\ell$  and  $F$  are the stage cost and the terminal cost, respectively. The stage cost  $\ell$  is assumed to satisfy  $\ell(z) \geq c|z|^2$ . Notice that if a terminal cost is to be included, it operates on the terminal state. Other ways of defining the cost function can be found in Maciejowski (1991).

The objective in MPC is to minimize the cost function (1) with respect to the control sequence  $\mathbf{u}$ , subjected to constraints on both state variables and the control sequence

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{R}^N} V(x(0), \mathbf{u}) \\ u(i) \in \mathbb{U}, 0 \leq i < N \\ x(i) \in \mathbb{X}, 0 \leq i \leq N \end{aligned} \quad (2)$$

where  $\mathbb{U}$  and  $\mathbb{X}$  are the sets of allowed control sequences and states, respectively. To be able to guarantee a unique optimum,  $\mathbb{U}$  is usually a convex, compact set and  $\mathbb{X}$  a convex, closed set, each with the origin included. When the optimal control sequence  $\mathbf{u}^0$  has been determined, only the first control input  $u^0(0)$  is applied to the plant  $P$  and the MPC procedure is repeated in the next sample.

Since the plant  $P$  is linear, if the stage cost is quadratic and the sets  $\mathbb{U}$  and  $\mathbb{X}$  are convex polytopes (i.e. constraints on the control sequence and state variables are linear), the optimization problem becomes convex and can be solved in a relatively efficient way. Hence it is often suitable to formulate the problem in such a way.

### 3. DYNAMIC MPC

The idea of Dynamic MPC, which is presented in this paper, is similar to traditional MPC. The main goal is to obtain an optimal control sequence which minimizes a certain cost function. The difference is that instead of directly calculate an optimal control sequence in each sample, an optimal dynamic controller is computed. Moreover, we do not optimize over the control values directly, instead these are parametrized via a dynamic compensator which in turn is linearly parametrized in a finite number of parameters. Our optimization goal is the same as before, i.e. to solve (2) in each step. We will give the details below.

#### 3.1 Formulation and Setup

Assuming that  $P$  is both stabilizable and detectable, there exist matrices  $K$  and  $L$  such that both  $\Phi - \Gamma_2 L$  and  $\Phi - K C_2$  are stable. By Youla, et al. (1976a,b); Boyd, et

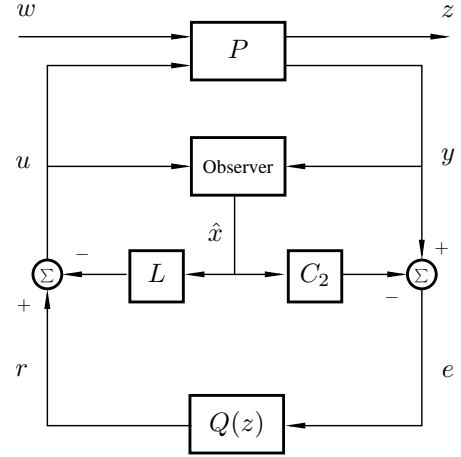


Fig. 1. Block diagram of the Youla-parametrization used in the paper

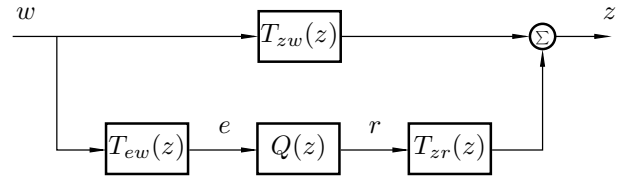


Fig. 2. Block diagram of the Youla-parametrization in the general form

al. (1991), it is known that the observer based nominal controller

$$\begin{aligned} \hat{x}(k+1) &= \Phi \hat{x}(k) + \Gamma_2 u(k) + K e(k) \\ u(k) &= r(k) - L \hat{x}(k) \\ e(k) &= y(k) - C_2 \hat{x}(k) \end{aligned} \quad (3)$$

combined with  $r = Q(z)e$  for a stable  $Q(z)$ , gives a stable system. If  $Q$  is viewed as parameter, this is what is called the Youla-parametrization or Q-parametrization. A diagram of the system can be found in figure 1.

An important condition in the Youla-parametrization is that the transfer function  $T_{er} \equiv 0$ , i.e. the transfer function from  $r$  to  $e$  is equal to zero. This condition is easily verified for the system described by  $P$  with the controller in (3). Using this condition, the transfer function of the system can be expressed as

$$G_{zw}(z) = T_{zw}(z) + T_{zr}(z)Q(z)T_{ew}(z)$$

where  $T_{zw}(z)$ ,  $T_{zr}(z)$  and  $T_{ew}(z)$  are the transfer functions of the system when  $Q(z)$  is removed. An illustrative diagram is found in figure 2.

Since the MPC procedure is performed in each step, and since the the initial state is changed in every step, care has to be taken when the initial state of the system does not equal zero. Consider the system

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 w(k) + B_2 r(k) \\ z(k) &= C_1 x(k) \\ e(k) &= C_2 x(k) \end{aligned}$$

If the system has the initial state  $x(0) \neq 0$  then, at time  $k$ , the outputs are

$$\begin{aligned} z(k) &= C_1 \{A^k x(0) + B_1 \mathbf{w}\} + C_1 B_2 \mathbf{r} \\ e(k) &= C_2 \{A^k x(0) + B_1 \mathbf{w}\} + C_2 B_2 \mathbf{r} \end{aligned} \quad (4)$$

where  $\mathcal{B}_1$  and  $\mathcal{B}_2$  are appropriate matrices and  $\mathbf{w} = [w(0), \dots, w(k-1)]^T$  and  $\mathbf{r} = [r(0), \dots, r(k-1)]^T$ . It follows from (4) the initial state  $x(0)$  should only be associated with one of the input transfer functions. Since  $T_{er} \equiv 0$ , the initial state must be associated with the transfer function for  $w$ , i.e. the transfer functions  $T_{zw}(z)$  and  $T_{ew}(z)$ .

Consider the plant  $P$  with the controller (3) combined with a  $Q(z)$  given by

$$Q(z) = q_0 + q_1 z^{-1} + \dots + q_{N-1} z^{-(N-1)}$$

where  $q_j \in \mathbb{R}^{n_u \times n_y}$ . It is clear that the closed loop is affine in the parameters  $q_j$ . In fact we shall show in section 5 that given an initial condition  $x(0)$ , the output at time  $i$  can be written as

$$z(i) = t(i) + h(i)\mathbf{q} \quad (5)$$

where  $\mathbf{q}$  contains all the parameters in the filter  $Q(z)$ . Consider now the following optimization problem

$$\begin{aligned} \min_{\mathbf{q}} V(x(0), \mathbf{q}) \\ u(i) \in \mathbb{U}, 0 \leq i < N \\ x(i) \in \mathbb{X}, 0 \leq i \leq N \end{aligned} \quad (6)$$

where the cost function is given by

$$V(x(0), \mathbf{q}) = \sum_{i=0}^{N-1} \ell(z(i)) + F(x(N)) \quad (7)$$

and each  $z(i)$  is given by (5).

*Remark 1.* As mentioned, we consider full state feedback. Further work involves analysis of the output feedback case.

### 3.2 Stability

From an optimization point of view the two problems (6) and (2) are similar, in particular in the standard convex MPC formulation they are equally easy to solve. Moreover, these two problems are equivalent from a control point of view

*Theorem 2.* Let  $V_d^*(x(0))$  and  $u_d^*(i)$  be the optimal cost and input trajectory corresponding to problem (6), let  $V_s^*(x(0))$  and  $u_s^*(i)$  be the optimal solution to (2). Then  $V_d^*(x(0)) = V_s^*(x(0))$ , and  $u_d^*(k) = u_s^*(k)$ ,  $0 \leq k \leq N-1$

**Proof.** Let  $x_s^*(k)$ ,  $0 \leq k \leq N$  be the states of  $P$  corresponding to the input sequence  $\mathbf{u}^* = (u_s^*(0), \dots, u_s^*(N-1))$ . What has to be shown is that there exists a

$$Q^*(z) = q_0^* + q_1^* z^{-1} + \dots + q_{N-1}^* z^{-(N-1)}$$

such that the system with the controller described by (3) also produces the input sequence  $\mathbf{u}^*$ , since if the sequence is the optimum of (2) then  $Q^*(z)$  must be the optimum of (6).

Let  $\tilde{x}^*(k) = x^*(k) - \hat{x}^*(k)$ , it can be shown that

$$e^*(k) = C_2 \tilde{x}^*(k) + D_{21} w(k)$$

Define  $e_k^* = (e^*(0), \dots, e^*(k))$  for  $0 \leq k < N$ . If we use  $u_d^*(k) = r^*(k) - L\hat{x}^*(k)$  and  $r^* = Q^*(z)e^*$ , it is easy to see that  $Q^*(z)$  can be chosen to satisfy

$$u_d^*(k) + L\hat{x}^*(k) = Q^*(z)e_k^*$$

*Remark 3.* An implication of the theorem is that stability is insured under the same conditions as for the traditional MPC with the same cost function and constraints. Such conditions can be found in e.g. Mayne, et al. (2000).

If constraints on the parameters of  $Q(z)$  are included, the system will be bounded-input bounded-output stable.

*Theorem 4.* Assume that for the plant  $P$  with initial state  $x(0)$ , the problem (6) combined with the constraints  $|q_i| \leq c_i$ ,  $0 \leq i \leq N-1$ , is feasible for all times  $k \geq 0$ . The resulting system when controlling  $P$  with Dynamic MPC is BIBO stable.

**Proof.** Let  $Q_k^0(z)$  be the optimal solution to (6) at time  $k$ . As will be seen in section 5

$$G_{zw}(z) = \left[ \begin{array}{c|c} A & B \\ \hline C_k & D_k \end{array} \right]$$

where  $A$  and  $B$  are constant for all times and  $C_k$  and  $D_k$  are linearly dependent on the parameters of  $Q_k^0(z)$ . Since the Youla-parametrization gives stable system, it is obvious that for bounded inputs  $w(k)$  the states  $x(k)$  will be bounded. By the restriction of the parameters of  $Q_k^0(z)$  it is also clear that  $\|C_k\|_2 \leq c$  and  $\|D_k\|_2 \leq d$  (for some  $c$  and  $d$ ), for all  $k \geq 0$ . This gives

$$\begin{aligned} \|z(k)\|_2 &\leq \|C_k\|_2 \cdot \|x(k)\|_2 + \|D_k\|_2 \cdot \|w(k)\|_2 \\ &\leq c\|x(k)\|_2 + d\|w(k)\|_2 \end{aligned}$$

Hence the system is BIBO stable.

Since the Dynamic MPC is equivalent to traditional MPC, when there are no computational delays, advantages of Dynamic MPC shows up when such delays are introduced. Assume that at some time  $k$ , the time required to find the optimal solution to the MPC problem is longer than the sample time. Not to leave the system uncontrolled, the optimal solution from the previous time instant  $k-1$  needs to be used. A straight-forward strategy in traditional MPC is to let the second control signal in the optimal control sequence be used as input to the system. This strategy is relying on open loop control, since the input does not rely on the current measure of the output. Dynamic MPC uses feedback to determine each control input. That is, even though no new optimal solution has been found yet, this strategy takes into account recent measured outputs when computing the next input. Hence, deviations of the measured outputs from the predicted outputs will be taken into account when the input is determined.

A FIR-filter  $Q(z)$  that is not ready at the time interval it was initially supposed to be optimal for, is in some sense outdated. Since the filter is optimized for the initial state at the time instant when the optimization began, it is most likely not optimal at the current state. To improve the performance of the system, we can update the complete controller by simulating it for the time when the optimization took place. Assume that the filter is delayed  $d$  samples and let  $u_{\text{meas}}(k)$  be the inputs to the plant and  $y_{\text{meas}}(k)$  be the measured outputs of the plant for the time during which the filter is being calculated, i.e.  $0 \leq k < d$ . Using the representation of the observer in (3), the update is performed according to

$$\begin{aligned} \hat{x}(k+1) &= \Phi \hat{x}(k) + \Gamma_2 u_{\text{meas}}(k) + K e(k) \\ x_Q(k+1) &= A_Q x_Q(k) + B_Q e(k) \\ e(k) &= y_{\text{meas}}(k) - C_2 \hat{x}(k) \end{aligned}$$

where  $A_Q$  and  $B_Q$  comes from the state space representation of  $Q(z)$ . Now  $\hat{x}(d)$  and  $x_Q(d)$  are the updated states that should be used to initialize the controller.

## 4. EXAMPLE

### 4.1 Double Integrator Example

To illustrate the presented ideas, first an example of the double integrator will be examined. The double integrator is

$$\begin{aligned}\dot{x} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\ y &= (1 \ 0) x\end{aligned}$$

which is discretized with a sample interval of 0.1s. The discrete model is set up according to figure 1, such that  $w$  contains the reference value,  $r$ , and  $z$  contains both  $y$  and the tracking error,  $r - y$ . The objective is to minimize the cost

$$\sum_{k=1}^N (r(k) - y(k))^2 + \sum_{k=1}^N p \cdot (\Delta u(k))^2$$

under constraints on the velocity,  $|x_2| \leq 0.1$ , and on the inputs,  $|u| \leq 0.3$ . The position  $y$  is to follow the reference trajectory  $r = 0.3$ .  $\Delta u(k)$  is the difference between the current and the previous input signal, i.e.  $\Delta u(k) = u(k) - u(k-1)$ . The prediction horizon  $N$  is set to 30 and the weight  $p = 0.3$ .

As seen by proposition 2, if there is no computational delay, the system will be equivalent to a system controlled by a traditional MPC controller. The response of the reference trajectory  $r = 0.3$  is found in 3(a). The corresponding velocity and control input are found in figures 3(b) and 3(c), respectively.

Now, suppose that there is a constant computational delay of 5 samples, i.e. if the optimization starts at time  $k$ , the optimum will not be determined and used until time  $k+5$ . Note that a constant computational delay is often not encountered. Depending on which constraints are active, the computations in the optimization takes different time to perform. But varying computational delays will not in general be different compared to constant delays, assuming that the controller does not know that the computational delay is not constant.

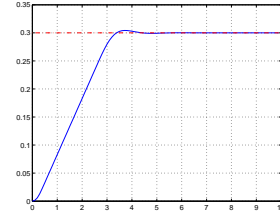
If traditional MPC is used to control the process, with the strategy that for an input sequence  $\mathbf{u}$  computed for time  $k$  (and is ready to use at  $k+5$ ), the input that is used at time  $k+i$  for  $5 \leq i \leq 9$ , is  $u(i)$ . The resulting system will be unstable and the response to the step  $r(k) = 0.3$  is presented in figure 4(a). The velocity and the applied control signal are found in figure 4(b) and 4(c).

Now, consider a Dynamic MPC controller. A controller that is computed during the time interval  $[k, k+5]$  is used to control the process during the interval  $[k+5, k+9]$ . Before the controller can be used, at time  $k+5$ , the controller states must be updated. This is done as explained in the last section.

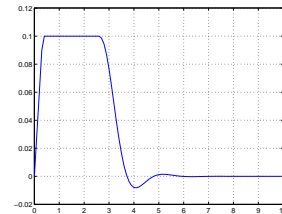
In figure 4(d) the trajectory of the system can be found when the reference of 0.3 is followed. Figure 4(e) shows the velocity of the system and figure 4(f) the input.

### 4.2 Buck Converter Example

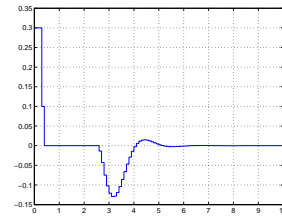
To give a more realistic example we consider a type of switch-mode DC-DC converter, called a Buck converter



(a) The position of the double integrator



(b) The velocity of the double integrator



(c) The input to the double integrator

Fig. 3. Plots of the position, velocity and input of the double integrator controlled with traditional or Dynamic MPC, without computational delay.

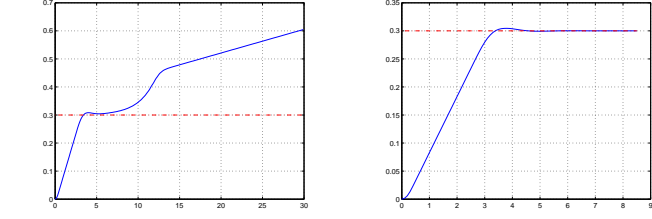
(see figure 5). Switch-mode converters are used in many applications such as any kind of power supplies. Due to the non-linear behaviour, these circuits can be challenging to control.

The output of the Buck converter is the voltage  $v_o$  and the inductor current. The control signal is defined by the duty cycle  $u$  as follows: At each sample time  $k$ ,  $u(k)$  takes a value in the interval  $[0, 1]$ , which represents the ratio of the interval  $[k, k+1]$  during which the switch in figure 5 is in position  $S_1$ . For the remaining part of the interval the switch is in position  $S_0$ . The convention is that the switch first is in position  $S_1$  and then in  $S_0$  in the sample interval. This results in a non-linear model for the circuit

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma(u(k)) \\ v_o(k) &= \frac{r_o}{r_o + r_c} [r_c \ 1] x(k)\end{aligned}$$

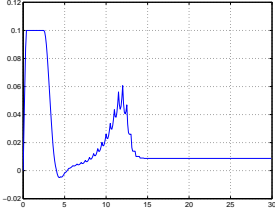
with the states  $x(k) = [i_\ell(k) \ v_c(k)]^T$ , where  $i_\ell(k)$  is the inductor current and  $v_c(k)$  the voltage over the capacitor. For further reference of the Buck converter, see e.g. Wernrud (2008).

The reference value of the output voltage  $v_o$  is set to be 20 V. Due to physical requirements the  $i_\ell$  has to be less than 2.3 A. The constraint on the duty cycle is  $0 \leq u \leq 0.9$ . We assume that a constant computational delay of 10 samples is present. The resulting voltage of the Buck converter

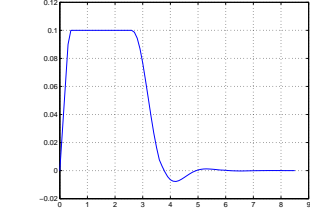


(a) The position of the double integrator (traditional MPC)

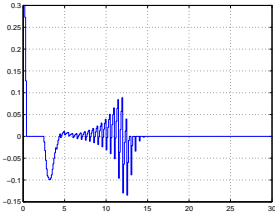
(d) The position of the double integrator (Dynamic MPC)



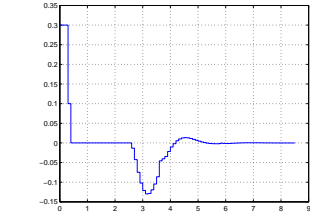
(b) The velocity of the double integrator (traditional MPC)



(e) The velocity of the double integrator (Dynamic MPC)



(c) The input to the double integrator (traditional MPC)



(f) The input to the double integrator (Dynamic MPC)

Fig. 4. Plots of the results of the double integrator when controlled with either traditional MPC or Dynamic MPC, in presence of a constant computational delay of 5.

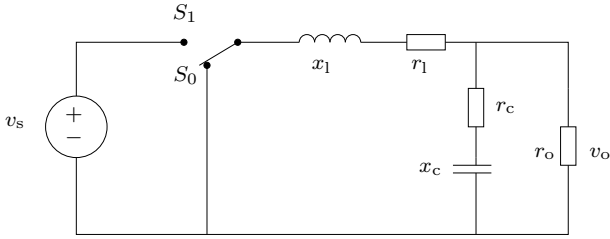


Fig. 5. Synchronous Buck converter.

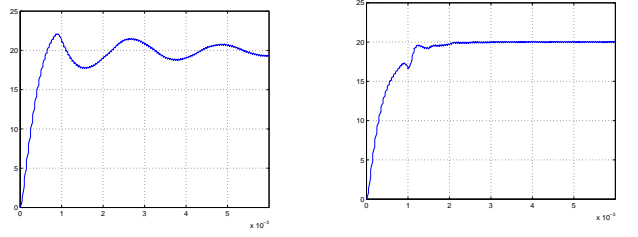
when controlled with either traditional MPC or Dynamic MPC, is shown in figure 6.

## 5. THE ON-LINE OPTIMIZATION PROBLEM

The purpose of this section is to give a brief review of how to exploit the Youla-parametrization for numerical computation, see Boyd, et al. (1991, 1988). In particular, we will show that the computational work required to solve the Dynamic MPC problem is similar to the work required in the traditional MPC formulation.

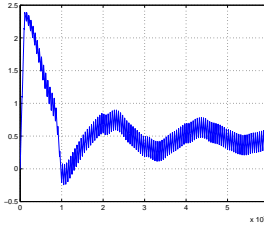
We have seen that the closed loop of any stabilizable linear system takes the form

$$G_{zw}(z) = T_{zw}(z) + T_{zr}(z)Q(z)T_{ew}(z)$$

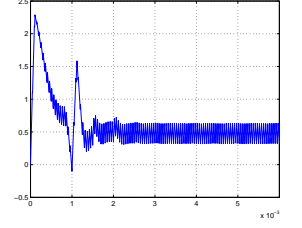


(a) Output voltage  $v_o$ , when traditional MPC is used.

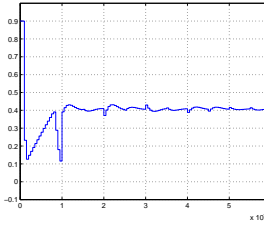
(d) Output voltage  $v_o$ , when Dynamic MPC is used.



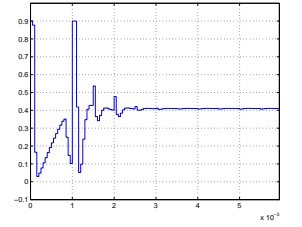
(b) Inductor current  $i_\ell$ , when traditional MPC is used.



(e) Inductor current  $i_\ell$ , when Dynamic MPC is used.



(c) Duty cycles  $u$ , when traditional MPC is used.



(f) Duty cycles  $u$ , when Dynamic MPC is used.

Fig. 6. The results when the Buck converter is controlled with traditional MPC and Dynamic MPC, respectively.

where  $T_{zw}(z)$ ,  $T_{zr}(z)$  and  $T_{ew}(z)$  are stable LTI-systems, depending only on the plant  $P$  and the nominal controller. Now consider the mapping between  $w_j$  and  $z_i$

$$\begin{aligned} G_{z_i w_j}(z) &= T_{z_i w_j}(z) + T_{z_i r}(z)Q(z)T_{e w_j}(z) \\ &= T_{z_i w_j}(z) + \sum_{k=1}^{n_u} \sum_{s=1}^{n_y} Q_{ks}(z)T_{z_i r_k}(z)T_{e_s w_j}(z) \end{aligned}$$

For the discussion in this section we may assume that  $n_u = 1$  and  $n_y = 1$ , it is trivial to go from this case to the general MIMO case. Moreover, we will drop the channel indices for notational simplicity. Accordingly, we denote the transfer function for any channel by

$$G_{zw}(z) = T_{zw}(z) + Q(z)T_{zr}(z)T_{ew}(z)$$

with scalar transfer functions. The only constraint on the free parameter  $Q$  is that it should be a stable rational LTI transfer functions, i.e.  $Q \in \mathcal{RH}_\infty$ . Thus the search space is infinite dimensional. To do numerical computations we need to restrict the search to a finite dimensional subspace. A simple choice is a FIR-base,  $z^{-l}$  for  $0 \leq l \leq N-1$ . Consider the candidate parametrization

$$Q(z) = \sum_{l=0}^{N-1} q_l z^{-l} = \begin{bmatrix} A_Q & B_Q \\ C_Q & D_Q \end{bmatrix}$$

where  $A_Q$  is the right shift matrix,  $B_Q$  is the first unit vector and

$$C_Q = [q_1 \ q_2 \ \dots \ q_{N-1}], \quad D_Q = q_0$$

Note that  $A_Q$  and  $B_Q$  are fixed. Let

$$T_{zr}(z)T_{ew}(z) = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

then we can take

$$\begin{aligned} H = Q(z)T_{zr}(z)T_{ew}(z) &= \begin{bmatrix} A_Q & B_Q C & B_Q D \\ A & \mathbf{0} & B \\ C_Q & D_Q C & D_Q \end{bmatrix} \\ &= \begin{bmatrix} \tilde{A} & \tilde{B} \\ C_Q & D_Q C \end{bmatrix} \end{aligned}$$

Finally we define

$$T_{zw}(z) = \begin{bmatrix} A_{zw} & B_{zw} \\ C_{zw} & D_{zw} \end{bmatrix}$$

Now, if the input to  $G_{zw}$  is  $w(0) \dots w(k)$ ,  $z(k)$  is given by

$$z(k) = t(k) + [C_Q \ D_Q C] \left( \tilde{A}^k x_0 + \sum_{p=0}^{k-1} \tilde{A}^{k-1-p} \tilde{B} w(p) \right) + D_Q$$

where  $t(k)$  is output of  $T_{zw}$ . We can write this more compact as

$$z(k) = t(k) + h(k)\mathbf{q}$$

where

$$\mathbf{q}^T = [D_Q \ C_Q] = [q_0 \ \dots \ q_{N-1}]$$

Also note that a realization of  $G_{zw}$  is given by

$$\begin{bmatrix} \tilde{A} & \mathbf{0} & \tilde{B} \\ \mathbf{0} & A_{zw} & B_{zw} \\ C_Q & D_Q C_p & C_{zw} \end{bmatrix} \begin{bmatrix} x \\ z \\ w \end{bmatrix}$$

### 5.1 Constraints in time-domain

Recall that the composition of a convex function and a linear function is a convex function. Let  $c_{ik}$  for  $1 \leq i \leq m$  be convex functions. Since the controlled output  $z(k)$  is a linear function of the decision variables  $\mathbf{q}$ , constraints of the form

$$c_{ik}(z(k)) \leq 0$$

are convex in  $\mathbf{q}$ . If the step cost  $l(z(k))$  is convex the finite horizon Dynamic MPC problem that we have defined in this paper can therefore be solved as a static finite dimensional convex optimization problem.

If the Dynamic MPC problem is formulated with only time-domain constraints, it is sufficient to use a FIR- filter to parametrize  $Q(z)$ . We consider only  $Q(z)$ 's such that

$$Q(z) \in \text{span}\{1, z^{-1}, \dots, z^{-(N-1)}\} \subset \mathcal{RH}_\infty$$

If we choose  $N$  larger then the optimization horizon  $N_h$ , non of the terms  $q_j z^{-j}$  with  $N_h \leq j \leq N-1$  will affect the achieved optimal cost nor the constraint satisfaction. On the other hand, if we include frequency-domain constraints the length of the FIR-filter will in general affect the outcome.

### 5.2 Constraints in frequency-domain

Again let  $c$  be a convex function. A robustness constraint of the form

$$c(G_{zw}(z)) \leq W(z), \quad z = e^{i\omega}, \omega \in [-\pi, \pi]$$

can be well approximated by restricting the last inequality to a finite set of points  $\omega_i$ . Each such constraint is convex

in  $\mathbf{q}$ . Some important frequency-domain constraint can be treated without the need of gridding. Consider for example the Bounded Real lemma. It says that if  $A$  is stable then

$$\|C(zI - A)^{-1}B + D\|_\infty < \gamma$$

if and only if

$$\begin{bmatrix} A^T X A - X & A^T X B & C^T \\ B^T X A & B^T X B - \gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0$$

Note that this inequality is linear in  $X, C$  and  $D$ . We have noted that in the realization of  $G_{zw}$  the decision variables enter only in the  $C$  and  $D$  matrices, thus frequency domain peak bounds results in convex constraints.

## 6. CONCLUSIONS

In this paper a new approach to model predictive control is developed, Dynamic MPC. It is shown that Dynamic MPC behaves as traditional MPC if there are no computational delays present. As computational delays are introduced, Dynamic MPC can take this into account and can in some cases stabilize a system where traditional MPC fails to do so.

## REFERENCES

- S.P. Boyd, C.H. Barratt. Linear Controller Design: Limits of Performance. Prentice Hall, 1991.
- S.P. Boyd, V. Balakrishnan, C.H. Barratt, N.M. Khraishi, X. Li, D.G. Meyer, S.A. Norman. A new CAD method and associated architectures for linear controllers. *IEEE Trans. on Automatic Control*, 33, pp. 268-283, 1988.
- C.E. Garcia, D.M. Prett, M. Morari. Model Predictive Control: Theory and Practice - A survey. *Automatica* 25, pp. 335-348, 1989.
- P. Goulart, E. Kerrigan, J.M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica* 42:4, pp. 523-533, 2006.
- D. Henriksson. Resource-Constrained Embedded Control and Computing Systems. PhD thesis ISRN LUTFD2/TFRT-1074-SE, Department of Automatic Control, Lund Institute of Technology, Sweden, January 2006.
- J. Lofberg. Approximations of closed-loop minimax MPC *IEEE Conference on Decision and Control*, 2003
- J.M. Maciejowski. Predictive control: with constraints Prentice Hall, 2002.
- D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica* 36, pp. 789-814, June, 2000.
- A. Wernrud. Approximate Dynamic Programming with Applications. PhD thesis ISRN LUTFD2/TFRT--1082-SE, Department of Automatic Control, Lund Institute of Technology, Sweden.
- D.C Youla, H.A. Jabr, J.J. Bongiorno. Modern Wiener-Hopf design of optimal controllers. Part 1: The single-input case. *IEEE Trans. on Automatic Control*, 21, February, 1976.
- D.C Youla, H.A. Jabr, J.J. Bongiorno. Modern Wiener-Hopf design of optimal controllers. Part 2: The multi-variable case. *IEEE Trans. on Automatic Control*, 21, June, 1976.