# A MULTIVARIABLE NONLINEAR ALGEBRAIC LOOP AS A QP WITH APPLICATIONS TO MPC

**A. Syaichu-Rohman[1], R.H. Middleton[2], M.M. Seron[3]**

*CIDAC, School of Electrical Engineering & Computer Science*
The University of Newcastle, University Drive, Callaghan, NSW 2308, Australia
[1]arohman@ee.newcastle.edu.au, [2]rick@ee.newcastle.edu.au, [3]seron@ee.newcastle.edu.au
Phone: +61-2-49216026, Fax:+61-2-49216993

**Keywords**: anti-windup, QP, MPC, algebraic loop, algorithm.

## Abstract

In this paper we consider an alternative approach to implementing Model Predictive Control (MPC). We show that solving a class of quadratic programming (QP) problems is equivalent to solving a class of well-posed non-linear algebraic loops. These algebraic loops are closely related to that found in static anti-windup synthesis. It therefore follows that certain classes of MPC can be implemented in a form that is a direct generalisation of the standard anti-windup structure. Implementations for classes of MPC are then derived for both regulator and tracking problems. Several alternative iterative algorithms for solving the algebraic loop are then presented. Some simulation results are also provided, with an example computational comparison to active set and interior point based MPC.

## 1 Introduction

In control design for linear systems, a saturation constraint on the control input is a very common problem that has long been studied. There are many approaches that have been proposed to incorporate actuation constraints in the control system design. Many of these approaches are based on either off-line or on-line optimisation. Two popular optimisation approaches are anti-windup design (e.g. [6,3]) and MPC (e.g. [8,2,5]).

MPC schemes typically use an on-line constrained optimisation. More recently, explicit MPC schemes which use off-line optimisation have been proposed [8,2,5] in an attempt to reduce the on-line computational burden. An attractive feature of MPC based controllers is their anticipative sense of both current and future constraints.

In an alternative stream of work, namely an anti-windup approach, the idea is to use the nominal linear controller and then to design, by off-line optimisation, a compensator for the deviation of the closed loop system from the linear behavior due to saturation on the current control input. For example, a static anti-windup scheme has been proposed in [6] where an LMI based synthesis to achieve an $L_2$ performance is employed. This static scheme can be seen as a special case of the dynamic anti-windup scheme [3] and the nonlinear algebraic loop, which the dynamic scheme may contain, can be avoided by restricting the LMI based synthesis to produce a strictly proper compensator. However, it has been shown in the example that the static scheme may give a better transient performance [6]. The static structure has also been included in [11] for this reason.

This work is motivated by connecting two observations from the literature: (i) the fact that MPC is a piece-wise linear control [8,2,5]; (ii) a static anti-windup approach is also a piecewise linear control [7]. Even though an MPC has, in general, an unknown algebraic structure, these facts raise the potential to generalize a static anti-windup scheme to an MPC scheme. One may also find a similar relation between a QP and a non-linear algebraic loop in [9] in which a QP problem is used to implement a ‘directionality compensator’.

In this paper, we show that a class of algebraic loops can be used to solve an on-line quadratic programming. Therefore, a class of MPC can be implemented in a block diagram as an alternative to MPC solutions [8,2,1,5] both for regulator and tracking problems.

The paper is organized as follows: Theorem 1 underlying the equivalence between a non-linear algebraic loop and a QP is explained in Section 2. In the following section, the application of Theorem 1 to MPC for regulator and tracking problems is presented. Some discussion on the robust realisation of the non-linear algebraic loop is also given here. Two simulation examples related to the regulator and tracking problems can be found in Section 4. Section 5 then concludes the paper followed by a list of references.

## 2 The Non-linear Algebraic Loop and Its QP implementation

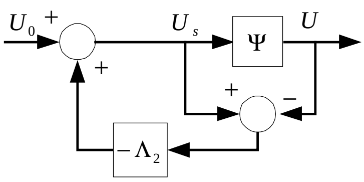


Fig.1 A Non-linear algebraic loop, [10]

A block diagram of a multivariable non-linear algebraic loop that can be found as a component of an anti-windup scheme may be drawn as shown in Fig. 1 where $\Psi(U_s)$ is a

$p$-dimensional diagonal saturation, that is $\Psi(U_s) := diag(\Psi_i(U_{s,i})) = U$ ; $U_s, U \in R^p$ , and for $i$=1,2, …, $p$

$$U_i = \Psi_i(U_{s,i}) = \begin{cases} U_{\min,i}; U_{s,i} < U_{\min,i} \\ U_i; U_{\min,i} \le U_{s,i} \le U_{\max,i} \\ U_{\max,i}; U_{s,i} > U_{\max,i} \end{cases} \quad (1)$$

Notice that if we define

$$M := (I + \Lambda_2)^{-1} \quad (2)$$

then by loop transformations Fig.1 is equivalent to Fig. 2.

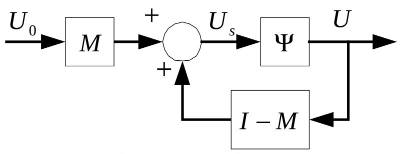


Fig.2 The QP implementation

We will now show that under certain conditions the loop in Fig. 2 gives a solution to a quadratic programme (QP). This is closely related to Section 4.2.5 in [9].

**Theorem 1:**
Consider the nonlinear multivariable algebraic loop in Fig.2. For $M$=$M^T$>0 and $U_0$ given, we then have

$$U = \Psi(U_s) \text{ , and} \quad (3)$$

$$U_s = U - M(U - U_0) \text{ , or}$$

$$M(U - U_0) = U - U_s \, . \quad (4)$$

Solving the algebraic loop in Fig.2, i.e. finding $U$ that satisfies (3)-(4), is equivalent to solving the following Quadratic Programming (QP):

$$U = \arg\min_{U \in U_{sat}} \{J\} \quad (5)$$

where $U_{sat}$ is the constraint set defined as

$$U_{sat} = \{U_{\min,i} \le U_i \le U_{\max,i}, i = 1,2,\dots,p\} \quad (6)$$

and

$$J = \tfrac{1}{2}(U - U_0)^T M (U - U_0) \, . \quad (7)$$

***Proof****:*
The first order Karush-Kuhn-Tucker (KKT) optimality conditions for the QP problem (5)-(7) are [2]:

$$M(U - U_0) - (\lambda_- - \lambda_+) = 0 \quad (8)$$

$$\lambda_{-,i}(U_{\min,i} - U_i) = 0 \text{ for } i = 1,2, \dots, p \quad (9)$$

$$\lambda_{+,i}(U_i - U_{\max,i}) = 0 \text{ for } i = 1,2, \dots, p \quad (10)$$

$$\lambda_+ \ge 0 \quad (11)$$

$$\lambda_- \ge 0 \quad (12)$$

$$U_{\min,i} \le U_i \le U_{\max,i} \text{ for } i = 1,2, \dots, p \quad (13)$$

where $\lambda_{+,i}$ and $\lambda_{-,i}$ are Lagrange multipliers and $\lambda_+ = \begin{bmatrix}\lambda_{+,1} & \lambda_{+,2} & \cdots & \lambda_{+,p}\end{bmatrix}^T$ and $\lambda_- = \begin{bmatrix}\lambda_{-,1} & \lambda_{-,2} & \cdots & \lambda_{-,p}\end{bmatrix}^T$ .

Condition (13) is clearly guaranteed by the definition of Ψ. Now rewrite (3) and (4) as

$$U = \Psi(U_s) = \Psi(U - M(U - U_0)) \, . \quad (14)$$

Then we have, for $i$=1,2, …, $p$

$$U_i - U_{s,i} = [M(U - U_0)]_i = \begin{cases} 0 & ; U_{\min,i} < U_i < U_{\max,i} \\ -\lambda_{+,i} & ; U_i = U_{\max,i} \\ \lambda_{-,i} & U_i = U_{\min,i} \end{cases} \quad (15)$$

where $\lambda_{+,i}$ , $\lambda_{-,i}$ for $i$=1,2, …, $p$ are non negative constants. In vector form, we can write

$$\lambda_- - \lambda_+ = M(U - U_0) = U - U_s \, . \quad (16)$$

It is easy to see that (15) and (16) are equivalent to (8)-(12), and hence $U$ in (14) is the optimizer for the QP problem and $\lambda_{+,i}$ , $\lambda_{-,i}$ defined in (15) are the Lagrange multipliers. The result then follows.□

**Remark 1: Feasibility and Well-posedness**

(1) The algebraic loop is well-posed if $M$=$M^T$>0.
(2) The QP problem is feasible with a unique solution if and only if the nonlinear algebraic loop is well-posed (uniqueness of solutions). □

**Remark 2: Lagrange Multipliers**
Implementation in Fig. 2 shows that the Lagrange multipliers given by equation (16) are the difference between the signals at the output and the input of the saturation block. □

**Remark 3**: **MPC-Antiwindup relation**
MPC, with control horizon one, is therefore equivalent to a special case of static-antiwindup, where $\Lambda_2$ is restricted to be symmetric, and $\Lambda_2$>-$I$. Note that by a small extension, Fig.2 also works with $M$=$D.Q$, where $Q$=$Q^T$>0 and $D$=diag($d_i$)>0, since diagonal scaling of the Lagrange multipliers does not alter the optimality conditions.

**Remark 4: MPC Applications**
A QP problem in an MPC can be implemented as a block diagram shown in Fig. 2 and the properties (e.g. stability) of the MPC are retained. The dimension of the diagonal saturation in the block diagram is equal to the dimension of $U$.□

# 3 Applications to MPC

### 3.1 MPC Regulator Problem

Consider a detectable plant

$$x_{k+1} = Ax_k + Bu_k \quad (17)$$

$$y_k = Cx_k \quad (18)$$

and an objective function

$$J = x_{k+N_y}^T \overline{Q}_{N_y} x_{k+N_y} + \sum_{t=0}^{N_y-1} x_{k+t}^T \overline{Q} x_{k+t} + \sum_{t=0}^{N_u-1} u_{k+t}^T \overline{R} u_{k+t} \quad (19)$$

where $x_k \in R^n$ , $y_k \in R^l$ , $u_k \in R^m$ , $x_{k+N_y}^T \overline{Q}_{N_y} x_{k+N_y}$ is a stability constraint [2], $N_y$= prediction horizon, $N_u$= control horizon, $N_u \le N_y$ , $\overline{Q} = \overline{Q}^T \ge 0$ , $\overline{R} = \overline{R}^T > 0$ . For simplicity, we take $N$=$N_u$=$N_y$ to define

$$X = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix}, \ U = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{bmatrix}, \qquad (20)$$

and have

$$X = \Phi x_k + \Gamma U$$

where $\Phi$ and $\Gamma$ are functions of $A$ and $B$ matrices, and $\overline{Q}_N = P$ is the solution of a discrete Riccati equation

$$A^T PA - P - A^T PB(\overline{R} + B^T PB)B^T PA + \overline{Q} = 0 \qquad (21)$$

Using MPC strategy, the problem is to design a controller for the plant by finding $U$ that minimises $J$ over the prediction horizon $N$, i.e. to find

$$U = \arg\min\{J\} \qquad (22)$$

subject to the constraint

$$u_{\min} \le u_{k+j} \le u_{\max}\ , j=0,1, \ldots, N\text{-}1 \qquad (23)$$

and use the first control vector in $U$ as the current control input for the plant

$$u_k = [I_m \quad 0 \quad \cdots \quad 0]U = E_1^T U\ . \qquad (24)$$

The above procedure is then repeated subsequently over the time.

Now, for prediction horizon $N$ we have the following equivalent problem:

Find $U$ to minimize

$$J_R = \tfrac{1}{2}(U + M^{-1}Fx_k)^T M(U + M^{-1}Fx_k) \qquad (25)$$

subject to

$$\mathbf{1}.u_{\min} \le U \le \mathbf{1}.u_{\max}\ . \qquad (26)$$

where

$$M = \Gamma^T Q \Gamma + R\ , \ F = \Gamma^T Q \Phi\ , \qquad (27)$$

$$\mathbf{1} = [I_m \quad I_m \quad \cdots \quad I_m]^T \qquad (28)$$

(array of $N$ identity matrices of $m$-size), and the matrices $Q$ and $R$ are functions of $P$, $\overline{Q}$ , and $\overline{R}$ matrices.

Notice that $J_R$ is readily in the form where the associated QP problem can be implemented using Theorem 1. By defining

$$U_0 = -M^{-1}Fx_k\ , \qquad (29)$$

the MPC for this regulator problem can be implemented in the block diagram shown below. In addition, the saturation levels of the $mN$-dimensional diagonal saturation $\Psi(.)$ have to be set up according to the constraint (26).

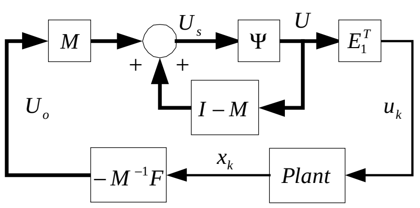


Fig. 3 MPC implementation of (25)-(29)

### 3.2 MPC Tracking Problem

Consider a detectable plant

$$x_{k+1} = \tilde{A}x_k + \tilde{B}u_k \qquad (30)$$

$$y_k = \tilde{C}x_k \qquad (31)$$

where $x_k \in R^n$, $y_k \in R^l$, $u_k \in R^m$ . The output $y_k$ is required to track an input reference $w_k \in R^l$ . To achieve zero tracking error, an integrator in the form of

$$u_k = u_{k-1} + \Delta u_k \qquad (32)$$

is augmented to the plant so that the the augmented plant can be written as

$$\overline{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = A\overline{x}_k + B.\Delta u_k \qquad (33)$$

$$y_k = C\overline{x}_k \qquad (34)$$

where $A$, $B$, and $C$ are appropriate matrices.

Define the objective function as

$$J = e_{k+N_y}^T \overline{Q}_N e_{k+N_y} + \sum_{t=0}^{N_y-1} e_{k+t}^T \overline{Q} e_{k+t} + \sum_{t=0}^{N_u-1} \Delta u_{k+t}^T \overline{R} \Delta u_{k+t} \qquad (35)$$

where $e_k = y_k - w_k$, $N_y$= prediction horizon, $N_u$= control horizon, $N_u \le N_y$, $\overline{Q} = \overline{Q}^T \ge 0$ and $\overline{R} = \overline{R}^T > 0$ .

As before, for simplicity we take $N = N_u = N_y$ to define

$$Y = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+N} \end{bmatrix}, W = \begin{bmatrix} w_{k+1} \\ w_{k+2} \\ \vdots \\ w_{k+N} \end{bmatrix}, \ U_d = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N-1} \end{bmatrix} \qquad (36)$$

and have

$$Y = \Phi \overline{x}_k + \Gamma U_d$$

where $\Phi$ and $\Gamma$ are functions of $A$, $B$, and $C$ matrices.

It is, however, often that the constraint on the amplitude of the control input is preferred than the rate of it. To this aim, one may use the following expression [4]:

$$U_d = DU - E_1 u_{k-1}\ , \qquad (37)$$

$$\text{with, } D = \begin{bmatrix} I_m & 0 & \cdots & 0 & 0 \\ -I_m & I_m & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_m & 0 \\ 0 & 0 & \cdots & -I_m & I_m \end{bmatrix} \qquad (38)$$

to get the following equivalent problem:

Find $U$ to minimise

$$J_T = \tfrac{1}{2}(U + M^{-1}F_X X)^T M(U + M^{-1}F_X X) \qquad (39)$$

subject to

$$\mathbf{1}.u_{\min} \le U \le \mathbf{1}.u_{\max}\ . \qquad (40)$$

where

$$X = [x_k^T \quad u_{k-1}^T \quad -W^T]^T\ , \qquad (41)$$

$$M = \overline{\Gamma}^T Q \overline{\Gamma} + D^T RD\ ,$$

$$F_X = \overline{\Gamma}^T Q \overline{\Phi} + D^T RG\ . \qquad (42)$$

The matrices $\overline{\Phi}$ , $\overline{\Gamma}$ , $Q$, $R$, and $G$ are functions of $A$, $B$, $C$, $D$, $E_1$, $\overline{Q}$ , and $\overline{R}$ matrices.

$J_T$ is now readily in the form in which the associated QP problem can be implemented using Theorem 1. Hence, by defining

$$U_0 = -M^{-1}F_X X = -M^{-1}[F \quad H]\begin{bmatrix}\bar{x}_k \\ -W\end{bmatrix} \qquad (43)$$

the MPC for this tracking problem can be implemented in the block diagram shown below. In addition, the saturation levels of the $mN$-dimensional diagonal saturation $\Psi(.)$ have to be set up according to the constraint (40).

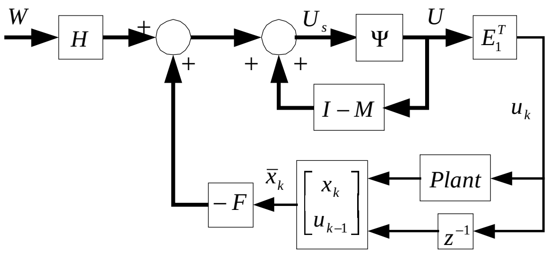


Fig. 4 MPC implementation of (39)-(43)

### 3.3 Algorithms For Solving The Algebraic Loop

Theorem 1 has shown that a QP problem with lower and upper bound constraint on the variables is equivalent to solving a multivariable algebraic loop involving a diagonal saturation with associated lower and upper level of saturations. Hence, implementing such algebraic loop in a discrete time is equivalent to solution of a QP.

Among many QP algorithms proposed in MPC problems, active-set methods are sometimes adopted in practice. Other algorithms, such as interior point methods, with stronger guaranteed performance results may also be used. Now, based on the fact explained in Theorem 1, we may derive other alternative algorithms from the feedback structure of the loop. The algorithms are simple to implement since they basically just iterate a variable of the loop until it converges.

**1. Algorithm-1 (original)**

Referring to Fig.2, we may have its associated iterative algorithms below.

$$U = \Psi(MU_0 + (I-M)U)) = \Psi(U - M(U-U_0)) \qquad (44)$$

Its iterative implementation may have the following form

$$U_{k+1} = \Psi(U_k - M(U_k - U_0)). \qquad (45)$$

Note that since $M = \nabla J$ , in the absence of saturation, this algorithm is essentially a steepest descent method with unity step length. It is well known that a steepest decent method is sensitive to a scaling. Hence, a scaling method, such as a constant or diagonal matrix scaling, may be used to improve the convergence rate of this algorithm. To analyze convergence, we note that $\Psi$ is a nonlinear operator with incremental gain $\leq 1$. Therefore, provided $0<M<2I$, the iteration defined by (45) converges exponentially to the single solution. This algorithm is also found in [9].

**2. Algorithm-2 (alternative-A)**

By loop transformation, we may get an alternative block diagram of the algebraic loop in Fig.2 as shown below.

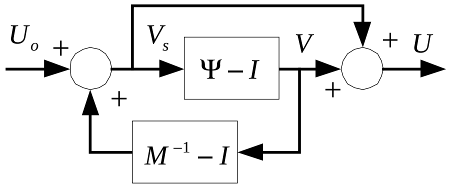


Fig.5 Block diagram of the Alternative-A

As before, to achieve a good convergence rate, a constant scaling for $M^{-1}$ may be employed. The iterative algorithm may be derived from this feedback structure as follows.
The loop equations, which are

$$V_s = U_0 + (M^{-1} - I)V \qquad (46)$$
$$V = \Psi_a(V_s)$$

may be implemented iteratively by

$$V_{k+1} = \Psi_a(U_0 + (M^{-1} - I)V_k) \qquad (47)$$

where

$$\Psi_a(x) := \Psi - I := \Psi(x) - x.$$

Provided $0<M^{-1}<2I$, that is $M>0.5I$, (47) converges exponentially fast. When the iteration converges, the solution of the loop is then obtained by

$$U^* = V^* + U_0 + (M^{-1} - I)V^*.$$

**3. Algorithm-3 (alternative-B)**

By loop transformation, we may get another alternative block diagram of the algebraic loop (see also [10]) in Fig.2 as shown below.

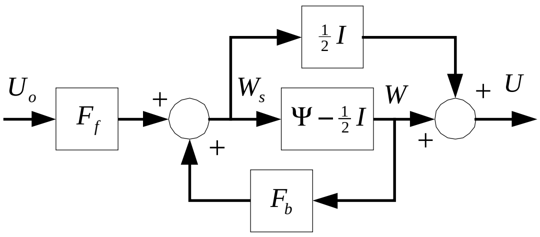


Fig.6 Block diagram of the Alternative-B

The iterative algorithm may be derived from this feedback structure as follows.
The loop equations, which are

$$W_s = F_f.U_0 + F_b.W \qquad (48)$$
$$W = \Psi_b(W_s)$$

may be implemented iteratively by

$$W_{k+1} = \Psi_b(F_f.U_0 + F_b.W_k) \qquad (49)$$

where $\Psi_b(W_s) := \Psi - 0.5I := \Psi(W_s) - 0.5W_s$ ,

$$F_f := 2(I+M)^{-1}M \ ; F_b := 2(I+M)^{-1}(I-M).$$

In this case, $\Psi_b$ is a nonlinear mapping with incremental gain $\leq 0.5$; and for any $M>0$, $F_b$ is a matrix with $\|F_b\|_{i2}<2$. Therefore, exponential convergence for (49) is guaranteed. When the iteration converges, the solution of the loop is then obtained by

$$U^* = W^* + \tfrac{1}{2}(F_f.U_0 + F_b.W^*).$$

## 4 Simulation Examples

Consider the continuous plant with the transfer function

$$P(s) = \frac{10}{100s+1}\begin{bmatrix} 4 & -5 \\ -3 & 4 \end{bmatrix}, \qquad (50)$$

which was used by many authors as an example in the context of anti-windup controller synthesis (e.g. [6]) and also recently in the context of MPC design in [2]. The discrete time model is obtained with sampling time $T$=1 sec. Referring to Section 3.3, the MPC tracking problem is set with

$$J = e_{k+N}^T \overline{Q}_N e_{k+N} + \sum_{t=0}^{N-1}(e_{k+t}^T \overline{Q} e_{k+t} + \Delta u_{t+k}^T \overline{R} \Delta u_{k+t}) \qquad (51)$$

$$\overline{Q} = diag(1{,}1)\,,\ \overline{R} = diag(0.1{,}0.1) \qquad (52)$$

$$w_k = [0.6 \quad 0.8]^T \text{ , and } N = N_y = N_u = 8 \qquad (53)$$

We look at the case where the constraint (over the horizon) is $-\mathbf{1} \le u_k \le \mathbf{1}$ with the objective function $J$ (51) and the MPC is implemented as shown in Fig. 4. and simulated in SIMULINK. Note that SIMULINK uses Newton method with rank-one-update for solving an algebraic loop. The simulation result is presented in Fig.7 with $u_{\min} = -\mathbf{1}$ for two different values of $u_{max}$.

To study the performance of four algorithms implementing a nonlinear algebraic loop, we do a discrete time simulation that is coded in MATLAB and run in a P3-800MHz PC. The simulation is now with varied input references until $t$=1500 sec and the results are shown in Fig.8 and Table 1.

In the table, $N$ is the control horizon, IP is for interior point method, *rcond* is the condition number of matrix $M$, and *flops* is the number of floating point operations required by each algorithm to get a solution within the specified tolerance. The worst case of convergence rate is provided in the last column, if it is applicable. Algorithm-1, 2 and 3 have also been scaled accordingly for their best theoretical computational performances. The infinity norm of the error between consecutive iterations is required to be less than or equal to a specified tolerance (which is $5.10^{-5}$) for stopping the iteration.

The table shows the performance of Algorithms-1, 2, and 3 relative to the active-set and interior point methods. It can be seen that the IP method is the best among the others in terms of peak flops. Although CPU time for MATLAB is not necessarily a good measure, the Algorithms-1, 2, or 3 is shown to be the fastest algorithm in terms of total simulation time.

Unlike the active set and IP methods, the performances of these three algorithms are dependent on the tolerance given. Hence, it needs a more elaborate performance analysis. In general, however, these algorithms show alternatives in which each algorithm needs a large number of iterations, but each iteration consists of simple operations. It is also noted that these iterative algorithms do not give quadratic convergence. However, they do allow guaranteed convergence rate, though in practice, this may be conservative.

## 5 Concluding Remarks

We have shown that a class of nonlinear algebraic loops that are commonly found in anti-windup schemes is equivalent to a QP. This observation helps unify the seemingly disparate fields of MPC and anti-windup. In particular, for a control horizon of one, MPC is a type of an anti-windup control, and for larger control horizons, anti-windup can be generalised to such case. Our formulation also leads to the implementation of the QP in MPC on a simple block diagram. Its application to MPC for regulator and tracking problems has also been presented. Several simple iterative algorithms equivalent to a constrained QP have been derived based on their associated feedback structure.

## Acknowledgement

We would like to thank Adrian Wills of the School of Electrical & Computer Science, University of Newcastle, Australia for providing his MATLAB codes of the interior point and active-set methods in the example.

## References

[1] Bemporad, A. and Filippi, C. (2001). Suboptimal Explicit MPC via Approximate Multiparametric Quadratic Programming, *Proceedings of the 40th Conf. On Decision & Control*, Orlando-USA.

[2] Bemporad, A., Morari, M., Dua, V., & Pistikopoulus, E.N. (2002). The Explicit linear quadratic regulator for constrained systems, *Automatica*, 38, 3-20.

[3] Grimm, G., Hatfield, J., Postlethwaite, I., Teel, A.R., Turner, M.C., and Zaccarian, L. (2001). Anti-windup: an LMI based synthesis, internal publication, Univ. of California, Santa Barbara, USA.

[4] Camacho, E.F. and Bordons, C. (1999). *Model Predictive Control*, London: Springer.

[5] Johansen, T.A., Petersen, I., & Slupphaug, O. (2002). Explicit sub-optimal linear quadratic regulation with state and input constraint. *Automatica*, 38, 1099-1111.

[6] Mulder, E.F., Kothare, M.V., & Morari, M. (2001). Multivariable anti-windup controller synthesis using linear matrix inequalities, *Automatica*, 37, 1407-1416.

[7] Romanchuk, B.G, and and Smith, M.C. (1996). Incremental Gain Analysis of Linear Systems with Bounded controls and Its Application to the Antiwindup Problem, *Proceedings of the 35th Conf. On Decision & Control*, Kobe-Japan, 2942-2947.

[8] Seron, M.M., De Doná, J.A., & Goodwin, G.C. (2000). Global Analytical Model Predictive Control with Input Constraint, *Proceedings of the 39th Conf. On Decision & Control*, Sydney-Australia.

[9] Soroush, M and Muske, K.R. (2000), Analytical Model Predictive Control, *Progress in Systems and Control Theory*, Vol.26, Basel-Switzerland:Birkhauser.

[10] Syaichu-Rohman, A. and Middleton, R.H. (2002). On The Robustness Of Multivariable Algebraic Loops With Sector Nonlinearities. *Proceedings of the 41st Conf. On Decision & Control*, Las Vegas-USA, 1054-1059

[11] Zaccarian, L., and Teel. A.R. (2002). A common framework for anti-windup, bumpless transfer and reliable design, *Automatica*, 38, 1735-1744.

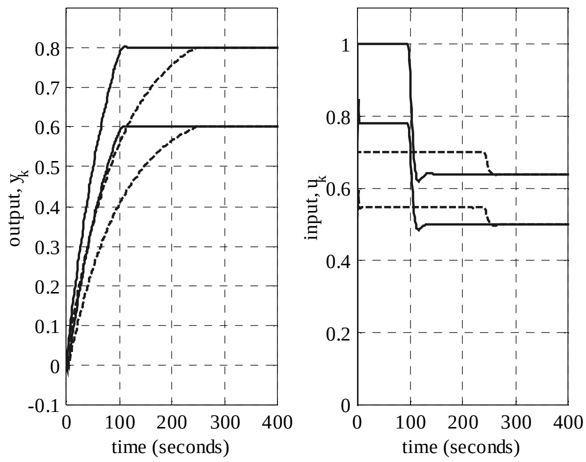


Fig. 7 Time responses for step references (solid: $u_{max}$=**1**; dashed: $u_{max}$=0.7x**1**)

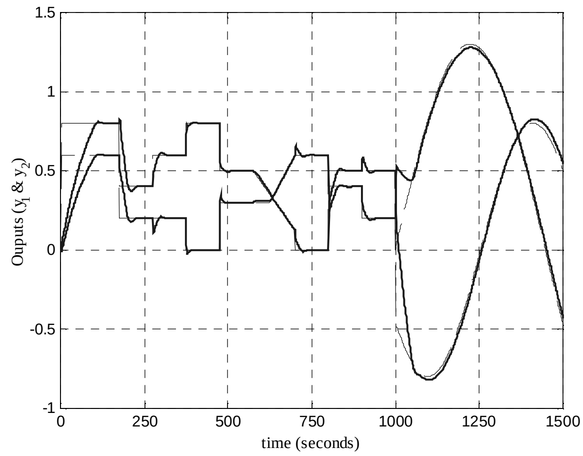


Fig.8 Output time responses for varied input references

| N (rcond) | Algorithm | Flops | | Sim.CPU Time (±) | Conv. Rate |
|---|---|---|---|---|---|
| | | Peak | Average | | |
| 33 ($2.05 \times 10^{-5}$) | IP | 977K | 301.8K | 207 s | n.a. |
| | Active-set | 1.32M | 979.9K | 50 s | n.a. |
| | Alg-1 | Stopped due to slow progress | | | 0.9999 |
| | Alg-2 | 9.18M | 292K | 47 s | ≈ 1 |
| | Alg-3 | 13.5M | 1.33M | 73 s | 0.9915 |
| 50 ($1.01 \times 10^{-5}$) | IP | 4.09M | 916.2K | 524 s | n.a. |
| | Active-set | 4.1M | 3.06M | 96 s | n.a. |
| | Alg-1 | Stopped due to slow progress | | | ≈ 1 |
| | Alg-2 | 21.4M | 705K | 75 s | ≈ 1 |
| | Alg-3 | 57.6M | 4.17M | 147 s | 0.9957 |
| 77 ($5.2 \times 10^{-6}$) | IP | 11.8M | 3.04M | 1738 s | n.a. |
| | Active-set | 13.9M | 10.5M | 237 s | n.a. |
| | Alg-1 | Stopped due to slow progress | | | ≈ 1 |
| | Alg-2 | 50.6M | 1.76M | 136 s | ≈ 1 |
| | Alg-3 | 247.8M | 14.6M | 425 s | 0.9978 |

Table 1: Performance comparison