An Antiwindup Interpretation of Input Constrained Model Predictive Control

Rick Middleton [a] Arief Syaichu-Rohman [b] María Seron [a] Adrian Wills [a]

[a] *Centre for Complex Dynamic Systems and Control*
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan 2308, NSW, Australia

[b] *Department of Electrical Engineering*
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia

Abstract

In this paper we consider the two seemingly disparate areas of linear input constrained Model Predictive Control (MPC) and Antiwindup control as frameworks to deal with actuator saturations. The MPC framework gives rise to a quadratic program whose solution satisfies the Karush-Kuhn-Tucker (KKT) conditions. Here we demonstrate that these KKT conditions form a set of implicit equations that may be represented as a block diagram containing an algebraic loop. Furthermore, this block diagram can be formed as a direct extension of antiwindup compensation. We also briefly discuss state constraint versions of the results as well as extended antiwindup type structures which exploit sparsity. The key overall result is that linear input constrained MPC is (globally) equivalent to antiwindup extended to include prediction of future control actions, their saturations, and compensation based on this.

Key words: model predictive control, antiwindup control, quadratic programming, actuator saturation, input constraints, state constraints.

1 Introduction

It has long been known (see for example [8], [21], [9]) that linear controller designs, applied to plants with significant actuator saturations can give very poor performance. One of the reasons for this performance degradation is often termed 'integrator windup' or 'reset windup'. This then leads to a large body of control research aimed at understanding this phenomena and mitigating the effects via 'antiwindup' schemes. Such schemes were initially designed in a somewhat ad-hoc manner (see for example [4], [7]). However, more recently analysis and synthesis of such schemes has been pursued for example using theory based on sector nonlinearities and involving linear matrix inequalities for the design (see for example [6], [12], [10]).

In what might appear to be a disjoint line of research, constrained optimisation approaches have been proposed as a means of dealing with actuator saturation. Typically, when appropriately formulated, finite dimensional optimisation problems arise resulting in the popular model predictive control (MPC) algorithms ([2], [5], [11], [13]). More recently, there has been increased interest in the relationship between these two approaches. For example, [3], has examined links between MPC and antiwindup for the single input case. In particular, it is shown that there are significant classes of systems for which in a non-trivial local domain, antiwindup and MPC give identical control actions. In this paper, we wish to pursue this line of research further, and show that for a broad class of systems, MPC for input constrained linear systems can be interpreted (globally) as an extended version of antiwindup control. This also allows a block diagram interpretation of MPC which may aid in the understanding of such algorithms.

The remainder of the paper proceeds as follows. In Section 2, we describe the system and the input constrained

⋆ Corresponding author María Seron.

Email addresses: Richard.Middleton@newcastle.edu.au (Rick Middleton), arief@lskk.ee.itb.ac.id (Arief Syaichu-Rohman), Maria.Seron@newcastle.edu.au (María Seron), Adrian.Wills@newcastle.edu.au (Adrian Wills).

MPC algorithm under consideration, including the formulation of the associated quadratic program. In Section 3 we show that the Karush-Kuhn-Tucker (KKT) conditions corresponding to this quadratic program can be represented as a block diagram. We also show that, by rearranging this block diagram, MPC is seen to be equivalent to a form of antiwindup control. In Section 4 we provide extensions of the results to include hard and soft state constraints and discuss a sparse structure formulation. In Section 5 we present some concluding remarks.

2 Preliminaries

We consider a linear time invariant discrete time plant with m inputs, state dimension n and ℓ outputs. The strictly proper plant transfer function matrix is denoted by $P(z)$ with an observable and controllable realisation:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

The plant input u_k is subject to a diagonal saturation constraint, which for simplicity we take to be symmetric and of unit amplitude

$$u_k \in \mathcal{U} := \{u \in \mathbb{R}^m : |u_i| \leq 1,\ i = 1 \dots m\} \tag{2}$$

We shall primarily be concerned with state observer based feedback regulation, though the results here extend readily to tracking problems using ideas such as those in [17].

2.1 A Class of Input Constrained MPC Algorithms

[1] We denote a state observer for the plant (1) by

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + L\left(y_k - \hat{y}_k\right) \\ \hat{y}_k &= C\hat{x}_k \end{aligned} \tag{3}$$

Note that in this case, the one step ahead state estimate, $\hat{x}_{k+1}$ depends only on current and past data $\{u_k, u_{k-1}, \dots\}$, $\{y_k, y_{k-1}, \dots\}$.

For simplicity we consider the control and prediction horizons for the MPC algorithm to be equal and denote this horizon by N. We further define, at the time instant k, future control actions $\hat{u}_{k+1|k}, \hat{u}_{k+2|k} \dots \hat{u}_{k+N|k}$ and based on these and the current state estimates, predicted future states

$$\begin{aligned} \hat{x}_{k+1|k} &= \hat{x}_{k+1} \\ \hat{x}_{k+\ell+1|k} &= A\hat{x}_{k+\ell|k} + B\hat{u}_{k+\ell|k}, \quad \ell = 1 \dots N \end{aligned} \tag{4}$$

[1] Note that for simplicity, the description here is for linear time invariant plants and cost functions, however, with further complexity in notation, the results generalise immediately to the linear time varying case.

Given control, state and final state weighting matrices, $0 < R = R^T \in \mathbb{R}^{m\times m}$, $0 \leq Q = Q^T \in \mathbb{R}^{n\times n}$, and $0 \leq P = P^T \in \mathbb{R}^{n\times n}$ respectively, we define the quadratic cost function

$$\begin{aligned} J_k\left(\hat{x}_{k+1}, \hat{u}_{k+1|k} \dots \hat{u}_{k+N|k}\right) &= \hat{x}^T_{k+N+1|k} P \hat{x}_{k+N+1|k} \\ &+ \sum_{\ell=1}^{N} \left(\hat{u}^T_{k+\ell|k} R \hat{u}_{k+\ell|k} + \hat{x}^T_{k+\ell|k} Q \hat{x}_{k+\ell|k}\right) \end{aligned} \tag{5}$$

where $\hat{x}_{k+\ell|k}$ is found by recursively using (4).

Note that frequently, the final state weighting P is chosen via the solution of an Algebraic Riccati Equation to give enhanced properties for the MPC algorithm (see for example [11]). For the case of input constraints only (2), the MPC algorithm then proceeds by solving a constrained optimisation problem, and assigning the actual control as the first element of the solution:

$$\begin{aligned} &\left\{u_{k+1|k}, u_{k+2|k} \dots u_{k+N|k}\right\} \\ &= \underset{\hat{u}_{k+\ell|k} \in \mathcal{U}, \ell=1\dots N}{\operatorname{argmin}} J_k\left(\hat{x}_{k+1}, \hat{u}_{k+1|k}, \hat{u}_{k+2|k} \dots \hat{u}_{k+N|k}\right) \\ &u_{k+1} = u_{k+1|k} \end{aligned} \tag{6}$$

2.2 Block Form MPC

It is well known (see for example [2],[13]) that the description of MPC in Section 2.1 can be written more succinctly by 'blocking' together various terms as follows. Define

$$\begin{aligned} \widehat{U}_k &= \begin{bmatrix} \hat{u}_{k+1|k} & \hat{u}_{k+2|k} & \dots & \hat{u}_{k+N|k} \end{bmatrix}^T \\ \mathcal{B} &= \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \\ \widehat{X}_k &= \begin{bmatrix} \hat{x}_{k+2|k} & \hat{x}_{k+3|k} & \dots & \hat{x}_{k+N+1|k} \end{bmatrix}^T \\ X_0 &= \begin{bmatrix} A\hat{x}_{k+1} & A^2\hat{x}_{k+1} & \dots & A^N\hat{x}_{k+1} \end{bmatrix}^T =: \mathcal{A}\hat{x}_{k+1} \end{aligned} \tag{7}$$

From the definitions in (7) we can rewrite the state equations, (4) in the compact notation:

$$\widehat{X}_k = X_0 + \mathcal{B}\widehat{U}_k \tag{8}$$

and the cost function, (5) may also be written more simply as

$$J_k\left(\hat{x}_{k+1}, \hat{U}_k\right) = \widehat{U}_k^T \mathcal{R} \widehat{U}_k + \widehat{X}_k^T \mathcal{Q} \widehat{X}_k + \hat{x}^T_{k+1} Q \hat{x}_{k+1} \tag{9}$$

where

$$\mathcal{R}=\text{blockdiag}\{R, R, \dots R\}$$
$$\mathcal{Q} = \text{blockdiag}\{Q, Q, \dots Q, P\}$$

Now denote by $\overline{U}_k$ the control which gives the unconstrained minimum of (9), obtained by minimising (9) subject to (8), that is:

$$\overline{U}_k = -\mathcal{K}X_0 \tag{10}$$

where the optimal linear gain $\mathcal{K}$ can be computed from

$$\begin{aligned}\mathcal{K} &= \mathcal{S}^{-1}\mathcal{B}^T\mathcal{Q}\\ \mathcal{S} &= \left(\mathcal{R}+\mathcal{B}^T\mathcal{Q}\mathcal{B}\right)\end{aligned} \tag{11}$$

Using (10) and (11) we can rewrite the cost function (9) as

$$\begin{aligned}J_k\left(\hat{x}_{k+1}, \hat{U}_k\right) &= \hat{U}_k^T\mathcal{R}\hat{U}_k + \hat{X}_k^T\mathcal{Q}\hat{X}_k + \hat{x}_{k+1}^T Q\hat{x}_{k+1}\\ &= \hat{U}_k^T\left(\mathcal{R}+\mathcal{B}^T\mathcal{Q}\mathcal{B}\right)\hat{U}_k + \hat{U}_k^T\mathcal{B}\mathcal{Q}X_0\\ &\quad + X_0^T\mathcal{Q}\mathcal{B}^T\hat{U}_k + X_0^T\mathcal{Q}X_0 + \hat{x}_{k+1}^T Q\hat{x}_{k+1}\\ &= \hat{U}_k^T\mathcal{S}\hat{U}_k + \hat{U}_k^T\mathcal{S}\mathcal{K}X_0 + X_0^T\mathcal{K}^T\mathcal{S}\hat{U}_k\\ &\quad + X_0^T\mathcal{Q}X_0 + \hat{x}_{k+1}^T Q\hat{x}_{k+1}\\ &= \left(\hat{U}_k - \bar{U}_k\right)^T\mathcal{S}\left(\hat{U}_k - \bar{U}_k\right)\\ &\quad + X_0^T\left(\mathcal{Q}^{-1}+\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^T\right)^{-1}X_0\\ &\quad + \hat{x}_{k+1}^T Q\hat{x}_{k+1}\end{aligned} \tag{12}$$

Clearly from (12), since the latter two terms are independent of $\hat{U}_k$ minimisation of J_k in (9) is equivalent to minimisation of J_k' defined as

$$J_k'\left(\hat{x}_{k+1}, \hat{U}_k\right) = \left(\hat{U}_k - \bar{U}_k\right)^T\mathcal{S}\left(\hat{U}_k - \bar{U}_k\right) \tag{13}$$

In other words, with the definition

$$U_k = \begin{bmatrix} u_{k+1|k} & u_{k+2|k} & \dots & u_{k+N|k}\end{bmatrix}^T$$

then the MPC problem defined in (6) is equivalent to the following quadratic program:

$$U_k = \underset{\hat{U}_k\in\mathcal{U}^N}{\text{argmin}}\left\{J_k'\left(\hat{x}_{k+1}, \hat{U}_k\right)\right\} \tag{14}$$

Equations (13), (14) define a quadratic program which requires solution during each sampling interval for the implementation of model predictive control. We next turn to consider how this quadratic program formulation may be extended to give a block diagram equivalent to the MPC algorithm.

3 Block Diagram Equivalent of MPC

3.1 Karush-Kuhn-Tucker (KKT) Conditions

We introduce the notation $\mathcal{I} = \begin{bmatrix} 1 & 1 & \dots & 1\end{bmatrix}^T$ as a vector of compatible dimension to U_k with all elements unity, and the notation $\mathcal{I} \geq U_k \geq -\mathcal{I}$ to denote the saturation constraints required (see (2)), that is, the inequalities hold elementwise. We further introduce (vector) Lagrange multipliers, Λ_+ and Λ_-, associated with the positive and negative saturations respectively. Then the KKT conditions (e.g. [1], [14]) associated with (13), (14) are:

$$\mathcal{S}\left(U_k - \bar{U}_k\right) - \left(\Lambda_- - \Lambda_+\right) = 0 \tag{15}$$
$$\Lambda_+^T\left(U_k - \mathcal{I}\right) = 0 \tag{16}$$
$$\Lambda_-^T\left(-U_k - \mathcal{I}\right) = 0 \tag{17}$$
$$\Lambda_+, \Lambda_- \geq 0 \tag{18}$$
$$\mathcal{I} \geq U_k \geq -\mathcal{I} \tag{19}$$

From (19) and the complementarity conditions (16) and (17), it follows that $\{\Lambda_{-_i} > 0\} \Rightarrow \{U_{k_i} = -1\} \Rightarrow \{U_{k_i} < +1\} \Rightarrow \{\Lambda_{+_i} = 0\}$. Conversely, $\{\Lambda_{+_i} > 0\} \Rightarrow \{\Lambda_{-_i} = 0\}$ and therefore we can, with no loss of information, replace the role of the two vectors of Lagrange multipliers, Λ_+ and Λ_- with the single vector:

$$\Lambda := \Lambda_+ - \Lambda_- \tag{20}$$

We then have the following result (see also [17]):

Lemma 1 *Given $\bar{U}_k$ suppose that U_k satisfies (15)-(19). Then there exists $\tilde{U}$ such that U_k and $\tilde{U}$ satisfy*

$$U_k = \Psi\left(\tilde{U}\right) \tag{21}$$
$$\tilde{U} = U_k - \mathcal{S}\left(U_k - \bar{U}_k\right) \tag{22}$$

where Ψ is the symmetric diagonal saturation

$$\Psi_i\left(\tilde{U}\right) := \begin{cases} +1 : \tilde{U}_i > 1\\ \tilde{U}_i \ : -1 \leq \tilde{U}_i \leq 1\\ -1 : \tilde{U}_i < -1\end{cases} \tag{23}$$

PROOF. Given U_k satisfying (15)-(19) and Λ as defined in (20), let

$$\tilde{U} := U_k + \Lambda \tag{24}$$

From (15) and (20) we have that $\Lambda = -\mathcal{S}\left(U_k - \bar{U}_k\right)$ and substituting this in (24) establishes (22).

From (16), (17), (19) and (24), it follows that

$$\{\Lambda_i > 0\} \Rightarrow \{U_{k_i} = 1\} \text{ and } \left\{\tilde{U}_i > 1\right\}$$
$$\{\Lambda_i = 0\} \Rightarrow \{-1 \leq U_{k_i} \leq 1\} \text{ and } \left\{-1 \leq \tilde{U}_i \leq 1\right\}$$
$$\{\Lambda_i < 0\} \Rightarrow \{U_{k_i} = -1\} \text{ and } \left\{\tilde{U}_i < -1\right\}$$

Hence U_k and $\tilde{U}$ satisfy (21), (23) and the result then follows. □

Lemma 1 shows that any solution of the KKT equations (15)-(19) is also a solution of the implicit equations (21)-(23). A converse form of this result is given next.

Lemma 2 *Given $\bar{U}_k$ suppose that we can find U_k and $\tilde{U}$ such that* (21)-(23) *hold. Then U_k solves the quadratic program (14).*

PROOF. Given (21) and (22) define Λ as

$$\Lambda = \tilde{U} - U_k \tag{25}$$

which from (22) can be rewritten as

$$\Lambda = -\mathcal{S}\left(U_k - \bar{U}_k\right) \tag{26}$$

We now reverse the combination of (20) to define (where the max operations are taken elementwise)

$$\Lambda_+ = \max\{\Lambda, 0\} \tag{27}$$
$$\Lambda_- = \max\{-\Lambda, 0\} \tag{28}$$

Then (27) , (28) in (26) establish (15). Also, from (27), $\{\Lambda_{+_i} > 0\} \Leftrightarrow \{\Lambda_i > 0\} \Leftrightarrow \left\{\tilde{U}_i > U_{k_i}\right\} \Leftrightarrow \left\{\tilde{U}_i > \Psi\left(\tilde{U}_i\right)\right\} \Leftrightarrow \left\{\tilde{U}_i > 1\right\}$. This then establishes the complementarity condition, (16). A similar line of reasoning gives $\{\Lambda_{-_i} > 0\} \Leftrightarrow \left\{-\tilde{U}_i > 1\right\}$ which establishes the second complementarity condition (17). The definitions (27) and (28) establish (18) and the definition of the saturation function, Ψ guarantees (19). It therefore follows that any simultaneous solution to (21) and (22) admits a unique definition of Lagrange multipliers, Λ_+, Λ_- that satisfy the KKT conditions (15)-(19) and therefore solve the quadratic program (14). □

3.2 Block Diagram MPC Equivalents

One of the key implications of Lemma 2 is that we can give a block diagram interpretation of the quadratic program. This is illustrated below in Figure 1 where E_1 is the 1st (block) elementary matrix, $E_1 = \left[\, I_{m\times m}\ 0_{m\times m}\ \ldots\ 0_{m\times m} \,\right]^T$.

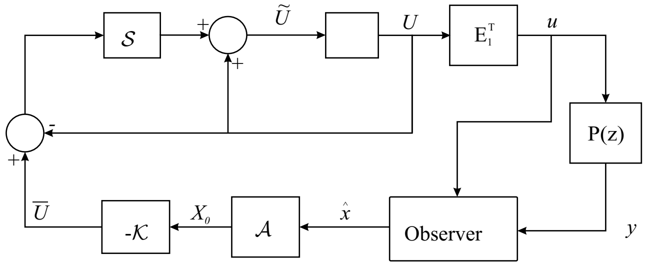

Fig. 1. Block diagram representation of Input Constrained Model Predictive Control

Simple block diagram manipulation of Figure 1 gives the equivalent form of Figure 2:

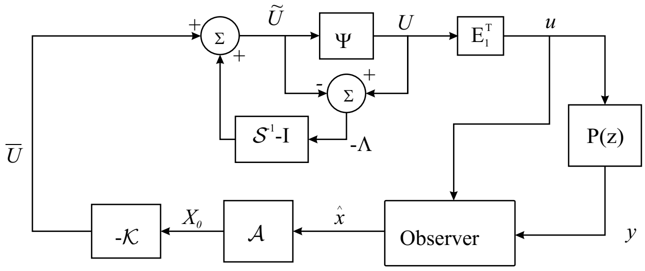

Fig. 2. Equivalent block diagram form of MPC

3.3 Antiwindup Formulation of Model Predictive Control

From Figure 2 and also noting that

$$\begin{aligned} U &= \mathcal{S}^{-1}\left(U - \tilde{U}\right) + \bar{U} \\ &= \mathcal{S}^{-1}\left(-\Lambda\right) + \bar{U} \end{aligned} \tag{29}$$

it is clear that another alternate form for Model Predictive Control is as shown in Figure 3.

Note that in this form, model predictive control can clearly be seen to be equivalent to a LTI controller $C(z)$ including a form of Antiwindup Control with the additional feature of taking into consideration predictions of both current and future unsaturated controls (in the stacked vector $\bar{U}$) together with feedback of the difference between current and future, saturated and pre-saturated controls (in the signal $U - \tilde{U}$). These observations lead directly to the following corollary.

Corollary 3 *Unit time horizon linear quadratic Model Predictive Control, with input saturation constraints only is globally equivalent to a linear time invariant controller together with static antiwindup compensation (see for example [12]).*

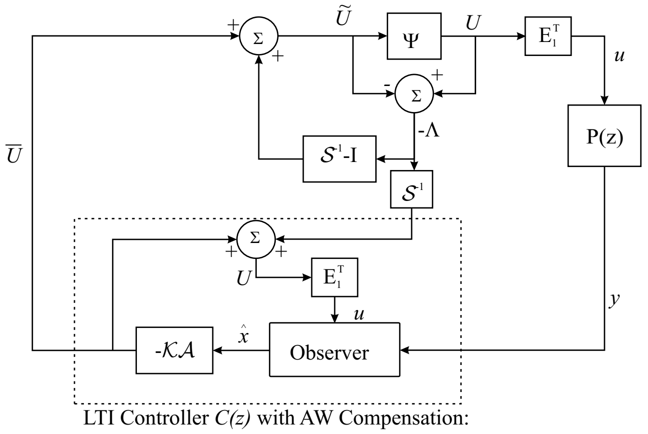

Fig. 3. Antiwindup form of Model Predictive Control

Note that the reverse conclusion, that antiwindup control is a special case of Model Predictive Control, cannot always be drawn, since the KKT conditions for MPC require that the matrix $\mathcal{S}$ be symmetric and positive definite. Positive definiteness is normally required to ensure well posedness of the algebraic loop in the antiwindup scheme, however, symmetry is not demanded by antiwindup schemes.

4 Extensions to the Basic Format

4.1 Incorporation of Hard State Constraints

One of the key benefits of model predictive control is that in addition to input constraints, and penalties on state deviations, it is also possible to take account of the need to impose limits on the permissible state deviations. The most common form of these are referred to as 'hard constraints' whereby in the optimisation, additional constraints are enforced. For linear inequality constraints, we represent these in block form as:

$$L\hat{X}_k \leq E \tag{30}$$

Then using (8) these constraints can be expressed as:

$$(L\mathcal{B})\,\hat{U}_k + (LX_0 - E) \leq 0 \tag{31}$$

Having transformed these constraints into linear inequality input constraints, we replace the previous input constraint region, $\mathcal{U}^N$ by the new, convex region:

$$\mathcal{U}_X(X_0) := \mathcal{U}^N \cap \left\{\hat{U}_k : (31)\right\} \tag{32}$$

For simplicity, we assume that this region is non-empty. In practice, some care must be taken to handle potential cases where large disturbances cause significant state deviations, and the intersection in (32) is empty. In the quadratic program formulation that follows, this creates an infeasible problem, that is, by definition the problem has no solution. A variety of techniques such as constraint priorities and constraint shedding (see for example [16,20]) have been introduced to deal with this troublesome issue, which we will not discuss further here.

Based on the assumption that $\mathcal{U}_X(X_0)$ is non-empty, the state and input constrained equivalent to (14) is:

$$U_k = \underset{\left\{\hat{U}_k \in \mathcal{U}_X(X_0)\right\}}{\operatorname{argmin}} \left\{J_k'\left(\hat{x}_{k+1}, \hat{U}_k\right)\right\} \tag{33}$$

To understand the hard state constraint situation further, we generalise the saturation function, Ψ, to an orthogonal projection $\Psi_{\mathcal{U}_X}$ onto the convex polytope $\mathcal{U}_X(X_0)$ with the definition:

$$\Psi_{\mathcal{U}_X}\left\{\tilde{U}\right\} := \underset{U \in \mathcal{U}_X(X_0)}{\operatorname{argmin}} \left\|U - \tilde{U}\right\|_2^2 \tag{34}$$

We now have the following generalisation to Lemma 2.

Lemma 4 *Given $\bar{U}_k$ suppose that we can find U_k and $\tilde{U}_k$ such that both*

$$U_k = \Psi_{\mathcal{U}_X}\left(\tilde{U}_k\right) \tag{35}$$

$$\tilde{U}_k = U_k - \mathcal{S}\left(U_k - \bar{U}_k\right) \tag{36}$$

where $\mathcal{S}$ is as defined previously in (11) and $\Psi_{\mathcal{U}_X}$ is the orthogonal projection defined in (34), then U_k solves the quadratic program (33).

PROOF. We represent the convex polytope by a set of linear inequality constraints:

$$\mathcal{U}_X(X_0) = \{U : \mathcal{L}U \leq \mathcal{E}\} \tag{37}$$

Then the orthogonal projection $\Psi_{\mathcal{U}_X}$ onto the convex region $\mathcal{U}_X(X_0)$ can be represented by the quadratic program

$$\Psi_{\mathcal{U}_X}\left(\tilde{U}_k\right) = \underset{\{U_k : \mathcal{L}U_k \leq \mathcal{E}\}}{\operatorname{argmin}} \left\|U_k - \tilde{U}_k\right\|_2^2 \tag{38}$$

The KKT conditions associated with (38) are:

$$U_k - \tilde{U}_k + \mathcal{L}^T\Lambda = 0 \tag{39}$$

$$\Lambda^T\left(\mathcal{L}U_k - \mathcal{E}\right) = 0 \tag{40}$$

$$\Lambda \geq 0 \tag{41}$$

$$\mathcal{L}U_k - \mathcal{E} \leq 0 \tag{42}$$

From (36) and (39) it follows that

$$\mathcal{S}\left(U_k - \bar{U}_k\right) + \mathcal{L}^T \Lambda = 0 \tag{43}$$

and equations (40)-(42) and (43) are precisely the KKT conditions for the quadratic program associated with (33). □

We therefore see that with state constraints, the block diagram of Figure 1 can be immediately generalised for the state constraint case to that shown in Figure 4

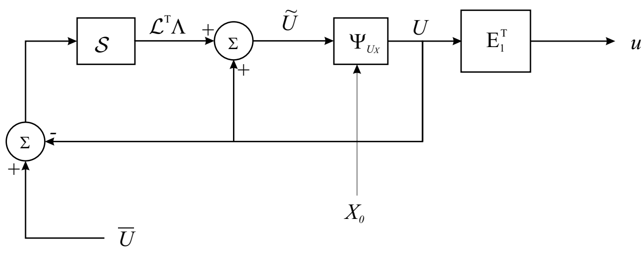

Fig. 4. Block Diagram for MPC with State Constraints

Note that in this case, although it is easy to draw the block diagram, it is difficult to gain further insights from this block diagram, since what was previously a saturation function, Ψ, is now in general an orthogonal projection, $\Psi_{\mathcal{U}_X}$, which requires solution of a quadratic program. This motivates us to consider alternate formulations for dealing with state constraints.

4.2 Incorporation of Soft State Constraints

Soft constraints, or closely allied penalty functions, are a common approach in optimisation to deal more easily with constraints. In such cases, a rigid requirement, is replaced by a sharply increasing penalty applied to violation of a constraint. This has the disadvantage that it does permit small excursions beyond an imposed limit, however, it does avoid some of the pitfalls of feasibility issues in Model Predictive Control ([22], [16]).

In our case, to consider the soft constraint version of (30) we introduce slack variables, ε_k, defined as follows:

$$\varepsilon_k = \max\{0, Z_k\} \tag{44}$$

$$Z_k := L\hat{X}_k - E = (L\mathcal{B})\,\hat{U}_k + (LX_0 - E) \tag{45}$$

where the max in (44) is the element by element maximum. We now modify the MPC cost function in (14) to include penalties on these additional variables as:

$$J_\varepsilon\left(\hat{x}_{k+1}, \hat{U}_k\right) = \frac{1}{2}\left(\hat{U}_k - \bar{U}_k\right)^T \mathcal{S}\left(\hat{U}_k - \bar{U}_k\right) + \frac{1}{2}\varepsilon_k^T \Omega \varepsilon_k \tag{46}$$

where Ω is a large positive definite weighting matrix, which for simplicity we take as being diagonal. The soft constrained MPC problem can then be written as:

$$U_k = \underset{\hat{U}_k \in \mathcal{U}^N}{\operatorname{argmin}} \left\{ J_\varepsilon\left(\hat{x}_{k+1}, \hat{U}_k\right) \right\} \tag{47}$$

This is a convex optimisation problem, since it can be shown that the additional soft constraints do not alter the convexity. In this case, the first KKT condition is, instead of (15):

$$\begin{aligned} 0 &= \mathcal{S}\left(U_k - \bar{U}_k\right) + \Lambda + \frac{\partial}{\partial \hat{U}_k}\left(\frac{1}{2}\varepsilon_k^T \Omega \varepsilon_k\right)\Big|_{\hat{U}_k = U_k} \\ &= \mathcal{S}\left(U_k - \bar{U}_k\right) + \Lambda + \left(\frac{\partial}{\partial \hat{U}_k}\left(Z_k^T \Omega\right)\Big|_{\hat{U}_k = U_k}\right)\varepsilon_k \end{aligned} \tag{48}$$

where the last line in (48) can be obtained by noting that

$$\begin{aligned} \frac{\partial}{\partial \hat{U}_{k_i}}\left(\frac{1}{2}\varepsilon_k^T \Omega \varepsilon_k\right) &= \frac{\partial}{\partial \hat{U}_{k_i}}\left(\sum_\ell \frac{1}{2}\varepsilon_{k_\ell}^2 \Omega_\ell\right) \\ &= \sum_\ell \Omega_\ell \frac{\partial}{\partial \hat{U}_{k_i}} \left\{ \begin{array}{c} \frac{1}{2} Z_{k_\ell}^2 : \varepsilon_{k_\ell} > 0 \\ 0 : \varepsilon_{k_\ell} = 0 \end{array} \right\} \\ &= \sum_\ell \Omega_\ell \varepsilon_{k_\ell} \frac{\partial Z_{k_\ell}}{\partial \hat{U}_{k_i}} \end{aligned}$$

Then using (45), (48) can be simplified to be:

$$0 = \mathcal{S}\left(U_k - \bar{U}_k\right) + \Lambda + \mathcal{B}^T L^T \Omega \varepsilon_k\big|_{\hat{U}_k = U_k} \tag{49}$$

The remaining KKT conditions for (47) are as previously in (16)-(19).

If we then define the 'switching' function:

$$f\left(\hat{X}\right) = \max\left\{0, L\hat{X} - E\right\} \tag{50}$$

then the solution to the soft constraint problem (47) can be represented by the block diagram shown in Figure 5.

Note that ε_k is zero, unless a soft state constraint has been exceeded. Therefore from Figure 5 we see that the incorporation of a soft state is equivalent to inclusion of an additional feedback term which is switched in when the state constraints are exceeded. This is conceptually similar to other feedback approaches to deal with state constraints such as the approach in [19].

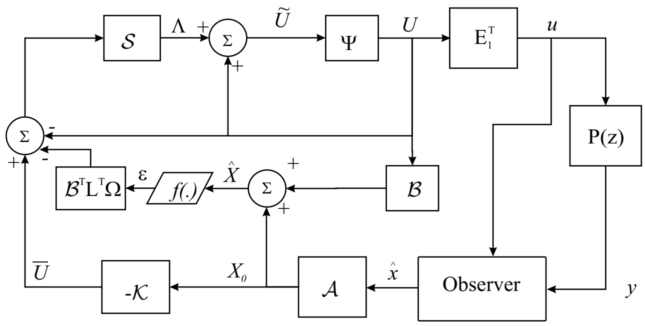

Fig. 5. Block Diagram of Model Predictive Control with Soft State Constraints

4.3 Sparse Structure Formulation

Note that in the forms suggested above in Figures 1 to 3, the complexity of the diagram is somewhat hidden due to the block nature of various signals and matrices. In fact, in the forms presented, since in general $\mathcal{S}$ is a full matrix of dimension $(Nm) \times (Nm)$ and this multiplies a vector, $\left(U_k - \bar{U}_k\right)$ of dimension (Nm) the overall complexity is $O\left(N^3 m^3\right)$, that is, it is cubic in the horizon of the MPC scheme and in the control dimension. However, it turns out that alternate formulations can be derived that exploit some sparsity properties of the underlying problem. In some cases, particularly if the control dimension and horizon are large, and the state dimension is not large, these sparse formulations yield improved complexity properties. This formulation closely parallels that of [15] where more efficient MPC formulation may be obtained for some classes of problems by retaining an over parametrised form with equality constraints, rather than eliminating intermediate variables immediately.

We first introduce some new stacked state variables. The first of these, $\bar{X}_k$ is the state predictions assuming no control saturations occur, defined as:

$$\bar{X}_k = X_0 + \mathcal{B}\bar{U}_k \tag{51}$$

where X_0 is as defined in (7) and $\bar{U}_k$ is the unconstrained controls as defined in (10). We now define the difference between the saturated and unsaturated state predictions as:

$$\begin{aligned} \tilde{X}_k &= \hat{X}_k - \bar{X}_k \\ &= \mathcal{B}\left(\hat{U}_k - \bar{U}_k\right) = \mathcal{B}\tilde{U}_k \end{aligned} \tag{52}$$

The definition of $\mathcal{B}$ in (7) is not sparse. However, note that if we define $C_B = \text{blockdiag}\left\{ B \ B \ \dots \ B \right\}$ and

$$\Gamma = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ A & 0 & 0 & \dots & 0 \\ 0 & A & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & A & 0 \end{bmatrix} \tag{53}$$

then it can be shown that

$$\mathcal{B} = (I - \Gamma)^{-1} C_B \tag{54}$$

Therefore, (52) can be re-written in the sparse form:

$$\tilde{X}_k = \Gamma\tilde{X}_k + C_B\tilde{U}_k \tag{55}$$

Also note, that after some algebraic manipulations, it is possible to show that subject to the constraint (55), the earlier cost function in (13) can be re-written as:

$$\begin{aligned} J_k^{"}\left(\hat{x}_{k+1}, \hat{U}_k, \hat{X}_k\right) &= \left(\hat{U}_k - \bar{U}_k\right)^T \mathcal{R}\left(\hat{U}_k - \bar{U}_k\right) \\ &\quad + \left(\hat{X}_k - \bar{X}_k\right)^T \mathcal{Q}\left(\hat{X}_k - \bar{X}_k\right) \end{aligned} \tag{56}$$

and the input constrained MPC problem is equivalent to:

$$U_k = \underset{\left\{ \begin{array}{c} \hat{U}_k \in \mathcal{U}^N \\ \tilde{X}_k = \Gamma\tilde{X}_k + C_B\tilde{U}_k \end{array} \right\}}{\text{argmin}} \left\{ J_k^{"}\left(\hat{x}_{k+1}, \hat{U}_k, \hat{X}_k\right) \right\} \tag{57}$$

We then introduce additional Lagrange multipliers, $M^T = \left[\mu_1^T \ \mu_2^T \ \dots \ \mu_N^T \right]$ associated with the equality constraints in (55). The KKT conditions for this problem are then:

$$\begin{aligned} \mathcal{R}\left(U_k - \bar{U}_k\right) - (\Lambda_- - \Lambda_+) + C_B^T M &= 0 \\ \mathcal{Q}\left(X_k - \bar{X}_k\right) - M + \Gamma^T M &= 0 \\ \Lambda_+^T\left(U_k - \mathcal{I}\right) &= 0 \\ \Lambda_-^T\left(-U_k - \mathcal{I}\right) &= 0 \\ \\ \Lambda_+, \Lambda_- &\geq 0 \\ \mathcal{I} \geq U_k &\geq -\mathcal{I} \\ -\left(X_k - \bar{X}_k\right) + \Gamma\left(X_k - \bar{X}_k\right) + C_B\left(U_k - \bar{U}_k\right) &= 0 \end{aligned}$$

It further follows that these equations can be represented in block diagram form as shown in Figure 6 where for simplicity we have suppressed the calculation through straightforward linear algebra of the unconstrained control inputs, $\bar{u}_1, \bar{u}_2 \ldots \bar{u}_N$.

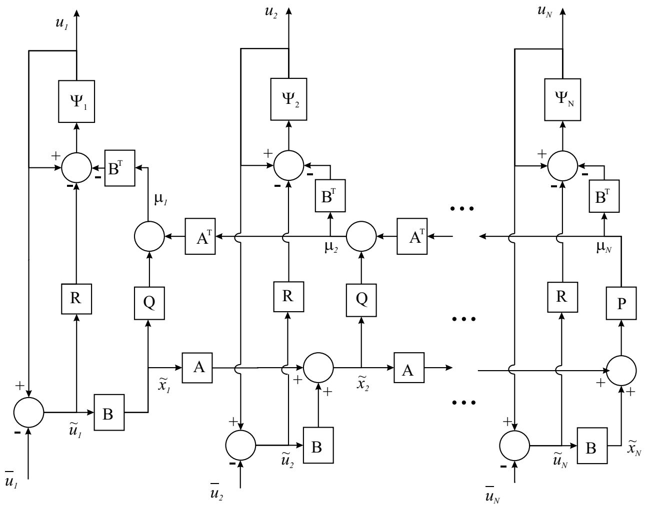

Fig. 6. Sparse block diagram representation of input constrained Model Predictive Control

The complexity of the structure shown in Figure 6 is of order $N\left(n^3+m^3\right)$ and therefore for large horizons, and where the state dimension is not significantly larger than the control dimension, the structure could be expected to offer significantly lower complexity than the form given earlier in Figure 3. Note also that the structure bears a close correlation to the dynamic programming structure of optimal control, where the μ variables in Figure 6 play the role of adjoint variables.

5 Conclusions

In this paper we have examined the KKT conditions for solution to a quadratic program that arises in linear MPC. These KKT conditions form a set of implicit, mainly linear equations. In the case of input constraints only, these implicit equations can be re-arranged in a form that is a direct extension of antiwindup control techniques. The extension required is to form not only the current control (with and without saturation) but also predictions of future control actions (with and without saturation) and to perform antiwindup on all current and predicted control actions. In the particular case of unit control horizon, MPC is globally equivalent to a special case of antiwindup control. We have also extended these results to state constrained MPC, and also considered sparse structure MPC which exhibits a close resemblance to the forward/backward formulations of discrete time dynamic programming.

The block diagram equivalents of MPC therefore offer insights into the nature of the feedback control, which is often obscured in the receding horizon optimisation framework. In addition, they inspire extension of antiwindup schemes (including LMI based designs) to predictive antiwindup schemes. The block diagram forms give rise to nonlinear multivariable algebraic loops, and therefore do not necessarily give direct implementations for the scheme, however, several algorithms for iteratively solving these loops have been proposed (see for example [18]). Such algorithms, typically exhibit a very low complexity in terms of coding, at a cost in terms of the number of iterations required for solution.

6 Acknowledgements

The authors would like to acknowledge helpful discussions during the formative stages of this work with Will Heath and José De Doná.

References

[1] S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge University Press, 2003.

[2] E. Camacho and C. Bordons, "Model Predictive Control", Springer, London, 1999.

[3] J.A. De Doná, G.C. Goodwin and M.M Seron, "Anti-Windup and Model Predictive Control: Reflections and Connections", European Journal of Control, V6, N5, 2000.

[4] J.C. Doyle, R.S. Smith, and D.F. Enns, "Control of plants with input saturation nonlinearities", In Proc. of the American Control Conference. Minneapolis, Minnesota, pp.1034-1039, 1987.

[5] C.E. Garcia, D.M. Prett and M. Morari, "Model Predictive Control: Theory and Practice - A survey", Automatica, V25, N3 pp.335-348, 1989.

[6] G. Grimm, J. Hatfield, I. Postlethwaite, A.R. Teel, M.C. Turner, and L. Zaccarian. "Anti-Windup for Stable Linear Systems With Input Saturation: An LMI-Based Synthesis," IEEE Trans. on Automatic Control V48, pp.1509-24, 2003.

[7] R. Hanus, and M. Kinnaert "Control of constrained multivariable systems using the conditioning technique.", In Proc. of the American Control Conference, Pittsburg, PA, pp.1712-1718, 1989.

[8] P. Kapasouris, and M. Athans "Design of feedback control systems for stable plants with saturating actuators". In Proc. of the 27th IEEE Conf. on Dec. & Control, Austin, Texas. pp.467-479, 1989.

[9] N. Kapoor, A.R. Teel, and P. Daoutidis, "An antiwindup design for linear systems with input saturation.", Automatica, V34, N5, pp.559-574, 1998.

[10] M.V. Kothare, P.J. Campo, M. Morari and C.N. Nett, "A Unified Framework for the study of Anti-Windup Designs", Automatica, V30, N12, pp.1869-1883, 1994.

[11] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: stability and optimality", Automatica, V36, pp.789-814, 2000.

[12] E.F. Mulder, M.V. Kothare and M. Morari, "Multivariable Anti-Windup synthesis using Linear Matrix Inequalities", Automatica, V37, N9, pp.1407-1416, 2001.

[13] J.R. Muske and J.B. Rawlings, “Model Predictive Control with Linear Models”, AIChE Journal, V39, N2, pp.262-287, 1993.

[14] J. Nocedal and S.J. Wright, “Numerical Optimization”, Springer-Verlag, New York, 1999.

[15] C. Rao, S. Wright, and J.B. Rawlings, “Application of interior-point methods to model predictive control”, Journal of Optimization Theory and Applications 99(3), 723-757, 1998.

[16] P. Scokaert and J.B. Rawlings, “Feasibility issues in linear model predictive control”, AiChE Journal, V45, N8, pp.1649-1659, 1999.

[17] A. Syaichu-Rohman, R.H. Middleton and M.M. Seron, “A multivariable nonlinear algebraic loop as a QP with application to MPC”, in Proc 7th European Control Conference, Cambridge, UK, September 2003.

[18] A. Syaichu-Rohman and R.H. Middleton, “Convergence study of some fixed point iteration QP algorithms”, Proc 43rd IEEE Conference on Decision and Control, Bahamas, Dec. 2004.

[19] M.C. Turner and I. Postlethwaite, ”Output violation compensation for systems with output constraint”, IEEE Transactions on Automatic Control, V47, N9, pp.1509-1525, 2002.

[20] J. Vada, O. Slupphaug, T.A. Johansen and B.A. Foss, “Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability”, Automatica, V37, pp.1835-1843, 2001.

[21] K.S. Walgama and J. Sternby “Conditioning technique for multiinput multioutput processes with input saturation.” IEE Proc. on Control. Theory and Applications V140, N4, pp.231-241, 1993.

[22] A.G. Wills and W.P. Heath, “An Exterior/Interior-point Approach to Infeasibility in Model Predictive Control”, Proc. IEEE Conference on Decision and Control, Maui, 2003