A Framework for Multivariable Algebraic Loops in Linear Anti-windup Implementations

Document Version
Accepted author manuscript

Published in:
Automatica

A Framework for Multivariable Algebraic Loops in Linear Anti-windup Implementations [⋆]

A. A. Adegbege [a], W. P. Heath [b]

[a] *Electrical and Computer Engineering, The College of New Jersey, Ewing, NJ 08628, USA.*

[b] *School of Electrical and Electronic Engineering, The University of Manchester, Manchester M13 9PL, UK.*

Abstract

This brief paper addresses the implementation and well-posedness aspects of multivariable algebraic loops which arise naturally in many anti-windup control schemes. Using the machinery of linear complementarity problems, a unified framework is developed for establishing well-posedness of such algebraic loops. Enforcing well-posedness is reduced to a linear matrix inequality feasibility problem that can be solved during the anti-windup design stage. Several existing anti-windup implementations appear as special cases of the unified framework presented in this brief paper.

Key words: Multivariable Algebraic loop; Anti-windup; Linear Complementarity Problem; Directionality Compensation; Constrained Control.

1 Introduction

Many constrained control problems involve dealing with algebraic loops comprising the feedback interconnection of a static nonlinearity and a linear term [11,15,29,20,4]. Such algebraic loops constitute delay-free circuits which may yield infinitely many solutions or no solution [31,9]. When a solution exists and it is unique, the algebraic loop is said to be well-posed (e.g. see [31,11,12,15,25,22,20]); the presence of an ill-posed algebraic loop can result in serious problems during practical implementation of control. Even when the algebraic loop is well-posed, numerical implementation may not be straightforward in real-time applications [28,12].

Recently, a number of ad-hoc strategies have been suggested to deal with specific cases of algebraic loop manifestations. In [28], the algebraic loop is broken by the introduction of a delay element. As shown, such strategies can result is serious performance degradation and may even induce closed-loop instability. As hinted in [11], a low-pass filter may be inserted into the loop to avoid

any implementation issue but at the expense of performance loss. In addition, the order of the resulting anti-windup is twice that of the unfiltered compensation, and the filter time constants have to be carefully chosen to avoid closed-loop instability. Implementation aspects of this strategy are discussed in [19]. Following a different approach, numerical algorithms have been suggested in [26,29,20].

In this brief paper, we present a framework for addressing both the well-posedness and the numerical implementation of static algebraic loops using the machinery of linear complementarity problems (LCP)[6]. Sufficient conditions for well-posedness are derived in terms of certain matrix classes which are well established in the mathematical programming literature. We show that many existing anti-windup designs correspond to particular implementations of the algebraic loop. Some preliminary results have been presented in [1,3]; here we provide formal proofs of the two main results (Proposition 5 and Proposition 7). In addition we include simulation results which demonstrate both the simplicity of on-line implementation of LCP solvers and the benefits offered by such a generalized framework. Finally, we comment on a simple procedure for the fast computation of algebraic loops. We discuss implementation issues, and in particular how the Lemke algorithm [5,6] can be tailored to various practical cases.

The notation employed throughout the brief paper is standard. However, following [6], a positive (semi) definite real matrix is not necessarily symmetric. For $x, y \in \mathbb{R}^m$, the notation $x > (\geq) y$ is used to mean $x_i > (\geq) y_i$ for $i = 1, \cdots, m$.

2 Problem Formulation

We consider the feedback interconnection of Fig. 1 which comprises a linear time invariant (LTI) subsystem denoted by P and a static algebraic loop subsystem. This block diagram is general, since H must be invertible to ensure well-posedness [31]; it provides a convenient structure for studying varieties of linear anti-windup systems. In this case, the subsystem P comprises the dynamics of the plant, the linear controller and the anti-windup compensation (for dynamic anti-windup). The LTI system is described by the state-space equations:

$$\dot{x} = Ax + B_1 w + B_2 \hat{u}, \tag{1a}$$
$$z = C_1 x + D_{11} w + D_{12} \hat{u}, \tag{1b}$$
$$v = C_2 x + D_{21} w + D_{22} \hat{u}, \tag{1c}$$

where $x \in \mathbb{R}^{n_p}$ is the state, $w \in \mathbb{R}^{n_w}$ is an external input, $z \in \mathbb{R}^{n_z}$ is an external output associated with some performance criterion, and $\hat{u}, v \in \mathbb{R}^m$ denote the second input and output of P. The algebraic loop subsystem is described by the following implicit relations:

$$\hat{u} = \text{sat}(u), \tag{2a}$$
$$u = Hv - (H - I)\hat{u}, \tag{2b}$$

where $H \in \mathbb{R}^{m \times m}$ is an invertible (not necessarily symmetric) matrix and $\text{sat}(u)$ is a decentralized saturation nonlinearity satisfying

$$\text{sat}(u) = \begin{bmatrix} \text{sat}_1(u_1) & \cdots & \text{sat}_m(u_m) \end{bmatrix}, \tag{3a}$$

where

$$\text{sat}_i(u_i) = \begin{cases} u_i^{max}; & u_i > u_i^{max}, \\ u_i; & u_i^{min} \leq u_i \leq u_i^{max}, \\ u_i^{min}; & u_i < u_i^{min}, \end{cases} \tag{3b}$$

with $u_i^{min} \leq 0$ and $u_i^{max} \geq 0$ for all $i = 1, \cdots, m$.

We assume that in the absence of saturation (i.e. when the algebraic loop subsystem is replaced with an identity), the unconstrained system is well-posed and has been designed to guarantee asymptotic stability. In this case, well-posedness implies that the matrix $(I - D_{22})^{-1}$ exists. Note that the feedthrough term from $\hat{u}$ to v in (1) may be subsumed into the algebraic loop to obtain
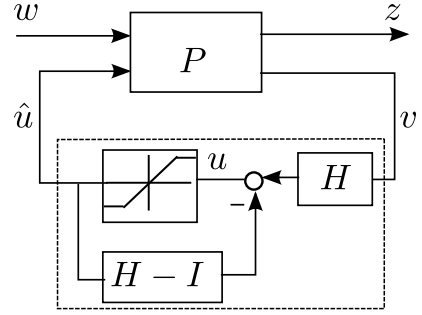

Fig. 1. Generalized Structure for Systems Incorporating Static Algebraic Loops

$$\begin{cases} \dot{x} &= Ax + B_1 w + B_2 \hat{u}, \\ z &= C_1 x + D_{11} w + D_{12} \hat{u}, \\ \tilde{v} &= \tilde{C}_2 x + \tilde{D}_{21} w, \end{cases} \tag{4a}$$

$$\begin{cases} \hat{u} &= \text{sat}(u), \\ u &= \tilde{H}\tilde{v} - (\tilde{H} - I)\hat{u}, \end{cases} \tag{4b}$$

where $\tilde{C}_2 = (I - D_{22})^{-1} C_2$, $\tilde{D}_{21} = (I - D_{22})^{-1} D_{21}$ and $\tilde{H} = H(I - D_{22})$. Thus well-posedness of the overall system reduces to that of the algebraic loop subsystem.

In this brief paper, we are concerned with the characterization and the practical implementation of all the different manifestations of algebraic loops of the form of (2) in linear anti-windup compensations. Using the framework of linear complementarity problems, we develop enforceable sufficient conditions under which a unique solution exists and we construct a simple procedure for real-time implementation of the algebraic loop.

3 Main Results

To state our main results, we give an alternative description of the algebraic loop which exploits the Karush-Kuhn Tucker (KKT) optimality condition of the saturation function [24]. We then characterize the class of matrix H for which the algebraic loop (2) is not only solvable but has a unique solution (i.e. well-posed).

Lemma 1 *Define $L \in \mathbb{R}^{m \times 2m}$ and $0 \leq b \in \mathbb{R}^{2m}$ as:*

$$L = [I \quad -I], \quad b = [(u^{max})^T \quad (-u^{min})^T]^T, \tag{5}$$

where u^{max} and u^{min} are the upper and lower bounds of (3) respectively. There exists a non-negative $\lambda \in \mathbb{R}^{2m}$ such that the input-output characteristics of the algebraic

loop subsystem (2) *satisfy the following conditions:*

$$H\hat{u} - Hv + L\lambda = 0, \tag{6a}$$

$$L^T\hat{u} - b \leq 0, \tag{6b}$$

$$\lambda^T(L^T\hat{u} - b) = 0. \tag{6c}$$

PROOF. See the appendix.

To test the feasibility of (6), we might seek vectors $\hat{u} \in \mathbb{R}^m$ (free) and $\lambda \in \mathbb{R}^{2m}$ (restricted) for a given input vector $v \in \mathbb{R}^m$, and fixed parameters L, b and H. Note that v and $\hat{u}$ are respectively the input and the output vectors of the algebraic loop subsystem (2). The algebraic loop description (6) comprises linear equations, linear inequalities and complementarity conditions and is generally termed a mixed LCP (mLCP) [32,6]. It is this description that we exploit in the following results. We describe in Definition 2 the class of **P**-matrices and the class of positive stable matrices [7], and in Definition 3 the Linear Complementarity Problem (LCP)[6].

Definition 2

(1) *A matrix $H \in \mathbb{R}^{m \times m}$ is a **P**-matrix if all of its principal minors are positive.*
(2) *A matrix $H \in \mathbb{R}^{m \times m}$ is said to be*
    (a) *diagonally dominant if*

$$|H_{ii}| \geq \sum_{j \neq i} |H_{ij}|, i = 1, \ldots, m. \tag{7}$$

    (b) *strictly diagonally dominant if strict inequality holds for all i.*
(3) *A matrix $H \in \mathbb{R}^{m \times m}$ is said to be*
    (a) *positive stable if there exists a symmetric positive definite matrix X such that HX is positive definite.*
    (b) *diagonally positive stable if there exists a positive-diagonal matrix X such that HX is positive definite.*

Definition 3
Given a vector $q \in \mathbb{R}^l$ and a matrix $Q \in \mathbb{R}^{l \times l}$, the LCP(q,Q) is to find a vector $\lambda \in \mathbb{R}^l$ such that

$$\lambda \geq 0, \ q + Q\lambda \geq 0, \tag{8a}$$

$$\lambda^T(q + Q\lambda) = 0. \tag{8b}$$

The LCP(q,Q) is said to be feasible if there exists a vector λ satisfying (8a). It is solvable if vector λ also satisfies (8b). The LCP (q,Q) is trivially solvable if $q \geq 0$.

The algebraic loop system (6) can be described as a variant of the standard LCP problem. Here, we are interested

in a class of monotone LCPs where Q is (not necessarily symmetric) positive semi-definite. For such a class, the LCP(q,Q) can have multiple solutions. Suppose Q can be partitioned into blocks of matrices as $Q = L^T H^{-1} L$. We have the following solvability results.

Lemma 4 *Let $H \in \mathbb{R}^{m \times m}$ be a P-matrix, and let $L \in \mathbb{R}^{m \times 2m}$ and $0 \leq b \in \mathbb{R}^{2m}$ be structured according to (5). Define $Q = L^T H^{-1} L$ and $q = b - L^T v$ where $v \in \mathbb{R}^m$ is arbitrary. The following statements hold:*

(1) *the LCP(q,Q) is feasible;*
(2) *if λ^* solves the LCP(q,Q), then $(q+Q\lambda^*)$ is unique, and λ^* is also unique provided $b > 0$.*

PROOF. See the appendix.

We now state our main well-posedness results.

Proposition 5 *Let $sat(\cdot) : \mathbb{R}^m \to \mathbb{R}^m$ be defined as (3) and suppose $L \in \mathbb{R}^{m \times 2m}$ and $0 \leq b \in \mathbb{R}^{2m}$ are structured according to (5). If $H \in \mathbb{R}^{m \times m}$ is a **P**-matrix, then the following statements hold:*

(1) *the algebraic loop formed by the feedback interconnection of $\hat{u} = sat(u)$ and $u = Hv - (H - I)\hat{u}$ is well-posed for any input vector $v \in \mathbb{R}^m$;*
(2) *for any two solutions $\hat{u}^1$ and $\hat{u}^2$ of the algebraic loop corresponding to the inputs v^1 and v^2, the solution map is Lipschitz continuous; i.e. there exist a positive constant c such that:*

$$(\hat{u}^1 - \hat{u}^2)^T H(v^1 - v^2) \geq c \|\hat{u}^1 - \hat{u}^2\|^2. \tag{9}$$

PROOF. See the appendix.

Remark 6 *Proposition 5 provides a useful test for checking well-posedness and for certifying the Lipschitz property of algebraic loops. It also implies that standard analysis tools [8,13] can be employed to assess the stability of feedback interconnections comprising non-conventional nonlinear structures such as the algebraic loop subsystem (2).*

In the sequel, we provide enforceable condition as a feasibility problem involving linear matrix inequalities (LMI). The result relies on the relationship between the class of diagonally positive stable matrices [7] and the class of **P**-matrices [6]:

Proposition 7 *Let $sat(\cdot) : \mathbb{R}^m \to \mathbb{R}^m$ be defined as (3) and suppose $L \in \mathbb{R}^{m \times 2m}$ and $0 \leq b \in \mathbb{R}^{2m}$ are structured according to (5). Let $H \in \mathbb{R}^{m \times m}$ be fixed.*

(1) If there exists a positive-diagonal matrix X such that

$$HX + XH^T \qquad (10)$$

is positive definite, the algebraic loop formed by the feedback interconnection of $\hat{u} = \text{sat}(u)$ and $u = Hv - (H - I)\hat{u}$ is well-posed for any input vector $v \in \mathbb{R}^m$.

(2) If there exists a positive-diagonal matrix X such that

$$H_{ii}X_i > \sum_{j \neq i} |H_{ij}|X_j, i = 1, \ldots, m, \qquad (11)$$

the algebraic loop formed by the feedback interconnection of $\hat{u} = \text{sat}(u)$ and $u = Hv - (H - I)\hat{u}$ is well-posed for any input vector $v \in \mathbb{R}^m$.

PROOF. See the appendix.

Remark 8 *The conditions in Items 1) and 2) of Proposition 7 are generalizations of the classes of (not necessarily symmetric) positive definite matrices and strictly row diagonally dominant matrices respectively. From item 1), we have that a diagonally positive stable matrix belongs to the class of **P**-matrices and from item 2), that a strictly row diagonally dominant matrix with positive diagonal entries is diagonally positive stable and hence belongs to the class of **P**-matrices.*

Remark 9 *Existing well-posedness results in the literature may be considered special cases of Proposition 7. For example, LMI (10) is exactly the condition in [28] for numerical robustness of the algebraic loop. Similar conditions are derived in [11,12] to guarantee solvability of such algebraic loops. In particular, the LMI condition of [12, Theorem 1] is feasible if and only if the LMI condition of Proposition 7, part (1) is feasible - i.e. the conditions for well-posedness are equivalent; while [12] considers a more general class of nonlinearity and quantifies the rank of an associated Lipschitz nonlinearity, Proposition 7, part (2) provides a stronger condition for well-posedness.*

On the other hand, [20] contains an explicit solution of the algebraic loop which requires that a certain matrix be strictly diagonally dominant. The method may be considered a special case of Proposition 7 (LMI (11)), but [20] provides no clear guideline on how the condition may be enforced during the anti-windup design stage.

4 Algebraic Loops in Anti-windup Control Implementations

The goal of anti-windup designs for linear systems with saturating actuators is to provide a mechanism for modifying the control actions during saturation so as to minimize its detrimental effects on closed-loop performance.

The aim of such modifications is mainly to recover as much as possible the linear performance or to provide graceful performance degradation when there are actuator saturations [33]. Let the plant be described by the following state-space realization:

$$\dot{x}_g = A_g x_g + B_{gw} w + B_{g\hat{u}} \hat{u}, \qquad (12a)$$
$$z = C_z x_g + D_{zw} w + D_{zu} \hat{u}, \qquad (12b)$$
$$y = C_y x_g + D_{yw} w + D_{yu} \hat{u}, \qquad (12c)$$

and the uncompensated controller by:

$$\dot{x}_k = A_k x_k + B_k(w - y), \qquad (13a)$$
$$u = C_k x_k + D_k(w - y), \qquad (13b)$$

where $x_g \in \mathbb{R}^{n_g}$ and $x_k \in \mathbb{R}^{n_k}$ are the states, $y \in \mathbb{R}^{n_y}$ is the plant output, and the controller output $u \in \mathbb{R}^m$ and the plant input $\hat{u} \in \mathbb{R}^m$ are related as $\hat{u} = \text{sat}(u)$. A standard assumption in anti-windup designs is that in the absence of saturation (i.e. when $u = \hat{u}$), the interconnection of (12) and (13) is well-posed and internally stable. Well-posedness of the unconstrained loop implies that the following matrices are well-defined:

$$\Delta_1 = (I + D_{yu}D_k)^{-1} \text{and } \Delta_2 = (I + D_k D_{yu})^{-1}. \quad (14)$$

Several well-known anti-windup schemes can be interpreted in terms of the algebraic loop structure of Fig. 1. The mLCP characterization in section 3 provides a framework for establishing well-posedness, and for easy and efficient online resolution of algebraic loops. Here, we highlight some manifestations of algebraic loops in linear anti-windup implementations and also note that the design freedom available through the algebraic loop feedback gain H is routinely used to equip the anti-windup for enhanced performance. We consider four types of anti-windup augmentations as follows:

4.1 Static Anti-windup [22]

The anti-windup structure of [18,22] is shown in Fig. 2 where the difference between the unsaturated control u and the saturated control $\hat{u}$ is fed through a compensating static filter Λ to generate two conditioning signals ξ_1 and ξ_2. Using the anti-windup augmentation $\xi_1 = \Lambda_1(u - \hat{u})$ and $\xi_2 = \Lambda_2(u - \hat{u})$, and the substitutions $H_1 = \Lambda_1(1 - \Lambda_2)^{-1}$ and $H_2 = (1 - \Lambda_2)^{-1}$, the augmented controller (i.e. the map from the tracking error $e = w - y$ to the plant input $\hat{u}$) can be decomposed into a linear system and an algebraic loop as follows:

$$\begin{cases} \dot{x}_k = (A_k + H_1 C_k)x_k + (B_k + H_1 D_k)e - H_1 \hat{u}, \\ v = C_k x_k + D_k e, \end{cases}$$
$$\begin{cases} \hat{u} = \text{sat}(u), \\ u = H_2 v + (I - H_2)\hat{u}, \end{cases} \qquad (15)$$

Fig. 2. The Unified Anti-windup Framework

Fig. 3. The Extended Weston-Postlethwaite Anti-windup Framework
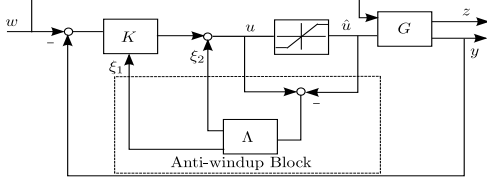
Note that the existence of $\Delta_2 = (I + D_k D_{yu})^{-1}$ guarantees the well-posedness of the linear portion and the algebraic loop subsystem is well-posed if H_2 is a **P**-matrix. In the generalized feedback structure of Fig. 1, we have

$$A = \begin{bmatrix} A_g & 0 \\ -\tilde{B}C_y & \tilde{A} \end{bmatrix}, B_1 = \begin{bmatrix} B_{gw} \\ \tilde{B}\tilde{D} \end{bmatrix}, B_2 = \begin{bmatrix} B_{gu} \\ -\tilde{B}D_{yu} - H_1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} C_z & 0 \end{bmatrix}, \begin{bmatrix} D_{11} & D_{12} \end{bmatrix} = \begin{bmatrix} D_{zw} & D_{zu} \end{bmatrix},$$

$$C_2 = \begin{bmatrix} -D_k C_y & C_k \end{bmatrix}, \begin{bmatrix} D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} D_k \tilde{D} & -D_k D_{yu} \end{bmatrix},$$

and $H = H_2$ where $\tilde{A} = A_k + H_1 C_k$, $\tilde{B} = B_k + H_1 D_k$ and $\tilde{D} = I - D_{yw}$. Note that here, the generalized plant has a feedthrough term $D_{22} = -D_k D_{yu}$. Subsuming this into the algebraic loop as for (4), we have $\tilde{C}_2 = (I + D_k D_{yu})^{-1} C_2$, $\tilde{D}_{21} = (I + D_k D_{yu})^{-1} D_{21}$ and $\tilde{H} = H_2(I + D_k D_{yu})$. It then follows from Proposition 5 that the overall interconnection is well-posed if $\tilde{H}$ is a **P**-matrix. Note that this condition is not more restrictive than the case of no feedthrough term. In fact, since by assumption the inverse of $(I + D_k D_{yu})$ exists, a sufficient condition for $\tilde{H}$ to be a **P**-matrix is that H_2 is a **P**-matrix (e.g. see [16, Theorem 3.2]).

Remark 10 *Observe that H_1 appears only in the linear subsystem of (15) and H_2 appears only in the algebraic loop. This separation between H_1 and H_2 suggests that the anti-windup design can be carried out in two stages: first, choose H_1 such that the overall closed-loop system is stable and then choose H_2 such that the algebraic loop is well-posed and to enhance closed-loop performance (see [2] for detailed discussion on this subject).*

4.2 Dynamic Anti-windup [11]

Here the anti-windup augmentation of Fig. 2 comprises a dynamic filter Λ described as:

$$\dot{x}_{aw} = \Lambda_1 x_{aw} + \Lambda_2 (u - \hat{u}), \tag{16a}$$

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} \Lambda_{31} \\ \Lambda_{32} \end{bmatrix} x_{aw} + \begin{bmatrix} \Lambda_{41} \\ \Lambda_{42} \end{bmatrix} (u - \hat{u}), \tag{16b}$$

where ξ_1 and ξ_2 are the conditioning signals, and $x_{aw} \in \mathbb{R}^{n_\lambda}$ is the state. Using the above augmentation, and the
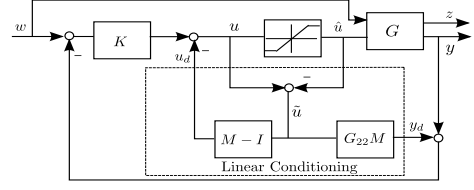
substitutions $H_1 = \Lambda_{41}(1 - \Lambda_{42})^{-1}$, $H_2 = (1 - \Lambda_{42})^{-1}$ and $H_3 = \Lambda_2(1 - \Lambda_{42})^{-1}$, the augmented controller can be decomposed into a linear system and an algebraic loop as follows:

$$\begin{cases} \begin{bmatrix} \dot{x}_k \\ \dot{x}_{aw} \end{bmatrix} = \begin{bmatrix} A_k + H_1 C_k & \Lambda_{31} + H_1 \Lambda_{32} \\ H_3 C_k & \Lambda_1 + H_3 \Lambda_{32} \end{bmatrix} \begin{bmatrix} x_k \\ x_{aw} \end{bmatrix} \\ \qquad + \begin{bmatrix} B_k + H_1 D_k \\ H_3 D_k \end{bmatrix} e - \begin{bmatrix} H_1 \\ H_3 \end{bmatrix} \hat{u}, \\ v = \begin{bmatrix} C_k & \Lambda_{32} \end{bmatrix} \begin{bmatrix} x_k \\ x_{aw} \end{bmatrix} + D_k e, \end{cases}$$

$$\begin{cases} \hat{u} = \text{sat}(u), \\ u = H_2 v + (I - H_2)\hat{u}, \end{cases} \tag{17}$$

where $e = w - y$ is the tracking error. In the generalized feedback structure of Fig. 1, we have

$$A = \begin{bmatrix} A_g & 0 & 0 \\ -\tilde{B}C_y & \tilde{A} & \Lambda_{31} + H_1 \Lambda_{32} \\ -H_3 D_k C_y & H_3 C_k & \Lambda_1 + H_3 \Lambda_{32} \end{bmatrix}, B_1 = \begin{bmatrix} B_{gw} \\ \tilde{B}\tilde{D} \\ H_3 D_k \tilde{D} \end{bmatrix},$$

$$B_2 = \begin{bmatrix} B_{gu} \\ -B_k D_{yu} - H_1 \Delta_2^{-1} \\ -H_3 \Delta_2^{-1} \end{bmatrix}, \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} C_z & 0 & 0 \\ -D_k C_y & C_k & \Lambda_{32} \end{bmatrix},$$

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} = \begin{bmatrix} D_{zw} & D_{zu} \\ D_k \tilde{D} & -D_k D_{yu} \end{bmatrix} \text{ and } H = H_2,$$

where $\tilde{A} = A_k + H_1 C_k$, $\tilde{B} = B_k + H_1 D_k$, $\tilde{D} = I - D_{yw}$ and Δ_2 is as defined in (14). Similar to the static anti-windup case, the interconnection is well-posed if $\tilde{H} = H_2(I + D_k D_{yu})$ is a **P**-matrix.

4.3 Extended Weston-Postlethwaite Dynamic Anti-windup [20]

The linear conditioning block in Fig. 3 acts as the anti-windup mechanism and contains M which comes from right coprime factorization of the plant transfer function

from $\hat{u}$ to y (denoted as $G_{22} = NM^{-1}$). The factorization is described as follows [20]:

$$\dot{x}_m = (A_g + B_{gu}F)x_m + B_{gu}E\tilde{u}, \tag{18a}$$
$$u_d = Fx_m + (E - I)\tilde{u}, \tag{18b}$$
$$y_d = (C_y + D_{yu}F)x_m + D_{yu}E\tilde{u}, \tag{18c}$$

where $x_m \in \mathbb{R}^{n_a}$ is the state of G_{22}, u_d and y_d are the conditioning signals, F is such that $A_g + B_{gu}F$ is Hurwitz and E is an invertible matrix. Using this compensation, the augmented controller can be decomposed into a linear system and an algebraic loop as follows:

$$\begin{cases} \begin{bmatrix} \dot{x}_k \\ \dot{x}_m \end{bmatrix} = \begin{bmatrix} A_k - B_k\Delta_1 D_{yu}C_k & -B_k\Delta_1 C_y \\ B_{gu}\Delta_2 C_k & A_g - B_{gu}\Delta_2 D_k C_y \end{bmatrix} \begin{bmatrix} x_k \\ x_m \end{bmatrix} \\ \qquad + \begin{bmatrix} B_k\Delta_1 \\ B_{gu}\Delta_2 D_k \end{bmatrix} e + \begin{bmatrix} B_k\Delta_1 D_{yu} \\ -B_{gu}\Delta_2 D_k \end{bmatrix} \hat{u}, \\ v = \begin{bmatrix} C_k & -(D_k C_y + (I + D_k D_{yu})F) \end{bmatrix} \begin{bmatrix} x_k \\ x_m \end{bmatrix} + D_k e, \end{cases}$$

$$\begin{cases} \hat{u} = \mathrm{sat}(u), \\ u = E^{-1}\Delta_2 v + (I - E^{-1}\Delta_2)\hat{u}, \end{cases} \tag{19}$$

where $e = w - y$ is the tracking error, and Δ_1 and Δ_2 are as defined in (14). Note that (19) takes the form of (2) and by assumption the inverses E^{-1}, Δ_1 and Δ_2 all exist. In the generalized feedback structure of Fig. 1, we have the following parameters:

$$A = \begin{bmatrix} A_g & 0 & 0 \\ -B_k\Delta_1 C_y & A_k - B_k\Delta_1 D_{yu}C_k & -B_k\Delta_1 C_y \\ -B_{gu}\Delta_2 D_k C_y & B_{gu}\Delta_2 C_k & A_g - B_{gu}\Delta_2 D_k C_y \end{bmatrix},$$

$$B_1 = \begin{bmatrix} B_{gw} \\ B_k\Delta_1\tilde{D} \\ B_{gu}\Delta_2 D_k\tilde{D} \end{bmatrix}, B_2 = \begin{bmatrix} B_{gu} \\ 0 \\ -B_{gu} \end{bmatrix},$$

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} C_z & 0 & 0 \\ -\Delta_2 D_k C_y & \Delta_2 C_k & -(\Delta_2 D_k C_y + F) \end{bmatrix},$$

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} D_{zw} & D_{zu} \\ \Delta_2 D_k\tilde{D} & 0 \end{bmatrix} \text{ and } H = E^{-1}.$$

where $\tilde{D} = I - D_{yw}$. It follows from Proposition 5 that the interconnection is well-posed if E^{-1} is a **P**-matrix, and provided Δ_1 and Δ_2 are well-defined.

4.4 Directionality compensation[23,27]

A class of anti-windup designs generally referred to as directionality compensation incorporates an artificial non-
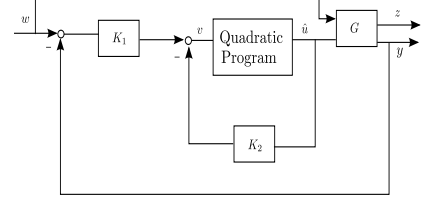

Fig. 4. Anti-windup with Symmetric Algebraic Loop expressed as Quadratic Program

linearity to deal with issues of input saturations and directionality in multivariable systems [23,27]. The artificial nonlinear typically takes the form of the following convex quadratic program (QP):

$$\arg\min_{\hat{u}} \frac{1}{2}\hat{u}^T H\hat{u} - \hat{u}^T Hv \text{ subject to } L^T\hat{u} \leq b, \tag{20}$$

where H is symmetric positive definite, $L = \begin{bmatrix} I & -I \end{bmatrix}$ and $b = [(u^{max})^T \ (-u^{min})^T]^T$. Note that the KKT optimality conditions for (20) are exactly the algebraic loop description given by (6) but with a symmetric positive definite matrix. In this case, we can make a stronger statement about the well-posedness of the algebraic loop.

Corollary 11 *Let $H \in \mathbb{R}^{m \times m}$ be a symmetric matrix and suppose the mLCP defined by (6) has a solution $(\hat{u}^*, \lambda^*)$. Then $\hat{u}^*$ is the unique global solution of the quadratic program (20) if and only if H is a **P**-matrix.*

PROOF. See the appendix.

Remark 12 *The implication of Corollary 11 is that when H is symmetric positive definite, the algebraic loop of Fig. 1 can be substituted by a QP whose unique solution coincides with the unique solution of the algebraic loop.*

The resulting anti-windup structure usually takes the form of Fig. 4 where the linear controller K has been factored into K_1 and K_2 and with K_2 wrapped around the quadratic program [2]. Depending on the factorization scheme adopted, different directionality compensation techniques can be realized. For example choosing K_1 and K_2 as the left coprime factors of K recovers the static anti-windup [22] and the factorization in section 4.1. However, choosing $K_1 = D_k$, $K_2 = K_1 K^{-1} - I$ and $H = D_k^{-T}D_k^{-1}$ recovers the conditioning scheme [23]. In this case, the augmented controller becomes:

$$\begin{cases} \dot{x}_k &= (A_k - B_k D_k^{-1}C_k)x_k + B_k D_k^{-1}\hat{u}, \\ v &= C_k x_k + D_k e, \\ \hat{u} = \mathrm{sat}(u), \\ u = D_k^{-T}D_k^{-1}v + (I - D_k^{-T}D_k^{-1})\hat{u}. \end{cases} \tag{21}$$

This also corresponds to choosing $H_1 = -B_k D_k^{-1}$ and $H_2 = D_k^{-T}D_k^{-1}$ in the static anti-windup scheme. Finally, choosing $K_1 = K$, $K_2 = 0$ and $H = \mathcal{C}^T\mathcal{C}$ where
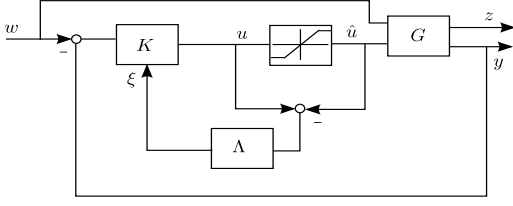
6

Fig. 5. One-Degree of Freedom Anti-windup Structure

$\mathcal{C}$ defines some directional characteristics of the plant, corresponds to the optimal directionality scheme of [27]. Standard design procedures are discussed in [2] and with robustness considerations in [4]. The quadratic programming formulation (20) has the advantage that it can be easily implemented online. There are many efficient algorithms such as the gradient projection methods [26] that are well-suited for its online solution. An explicit but suboptimal algorithm is discussed in [23].

4.5 Anti-windup without Algebraic loop [18]

Many anti-windup systems can be interpreted within the one-degree of freedom framework of Fig. 5 where the conditioning signal is injected only to the state of controller K or the Weston-Postlethwaite structure of Fig. 3 where the linear conditioning is governed by:

$$\dot{x}_m = (A_g + B_{gu}F)x_m + B_{gu}\tilde{u}, \qquad (22a)$$
$$u_d = Fx_m, \qquad (22b)$$
$$y_d = (C_y + D_{yu}F)x_m + D_{yu}\tilde{u}. \qquad (22c)$$

These anti-windup systems can be easily restructured into the generalized framework of Fig. 1 with H fixed as the identity i.e. $H = I$. For the static case where $\xi = \Lambda(u - \hat{u})$, the anti-windup corresponds to choosing $H_1 = \Lambda$ and $H_2 = I$ in the discussion of section 4.1. For the dynamic case where

$$\dot{x}_{aw} = \Lambda_1 x_{aw} + \Lambda_2(u - \hat{u}), \qquad (23a)$$
$$\xi = \Lambda_3 x_{aw} + \Lambda_4(u - \hat{u}), \qquad (23b)$$

the anti-windup corresponds to choosing $H_1 = \Lambda_4$, $H_2 = I$, $H_3 = \Lambda_2$, $\Lambda_3 = \Lambda_{31}$ and $\Lambda_{32} = \Lambda_{42} = 0$ in the discussion of section 4.2. Finally, using the augmentation (22), the anti-windup corresponds to choosing $D_{22} = -D_k D_{yu}$ and $E = I$ in the discussion of section 4.3. In all cases, there are no algebraic loops arising from the anti-windup augmentations and the existence of Δ_1 and Δ_2 guarantee well-posedness of the interconnections.

5 Algebraic Loop Computation

The mLCP problem can be solved using a number of algorithms including projected iterative methods [21], interior point algorithms [32] and pivoting schemes such as the Lemke algorithm [6,5]. The Lemke algorithm is

---

Algorithm 1 Lemke Algorithm

1: $q \leftarrow (b - L^T v)$, $Q \leftarrow L^T H^{-1} L$
2: **if** $q \geq 0$ **then**
3:    $\lambda \leftarrow 0$
4: **else**
5:    Create tableau $[I, -Q, -e, q]$
6:    $t \leftarrow \arg\min q_i$
7:    Pivot $< \lambda_o, s_t >$
8:    **while** λ_o is a basic variable **do**
9:      $r \leftarrow \arg\min q_i /$ driving variable
10:      Pivot $< \lambda_t, s_r >$
11:      driving variable $\leftarrow \lambda_r$
12:    **end while**
13:    $\lambda_i \leftarrow q_i$
14: **end if**
15: $\hat{u} = v - H^{-1} L\lambda$

---

particularly useful as it can be adapted for solving algebraic loops with either symmetric or asymmetric feedback gains. It is known to terminate at a unique solution, in a finite time, for the LCP (q, Q) when Q is a P-matrix [6]. In our case, Q is not necessarily a **P**-matrix. However, for the class of problems that we consider, Q is structured in such a way that the problem can be decoupled into smaller LCPs each with a **P**-matrix.

5.1 Lemke Algorithm for Algebraic Loops

The Lemke algorithm applies to the LCP problem LCP(q, Q). The first step involves introducing a non-negative slack variable s into (8) to obtain

$$s = Q\lambda + q, \qquad \lambda, s \geq 0, \qquad \lambda^T s = 0. \qquad (24)$$

Next a dummy variable λ_o is introduced to ensure a feasible start of the algorithm and such that

$$s = Q\lambda + e\lambda_o + q, \qquad \lambda, s \geq 0, \qquad \lambda^T s = 0, \quad (25)$$

where $e = [1 \cdots 1]^T \in \mathbb{R}^l$. The algorithm then follows a series of complementary pivoting steps and terminates at a solution when λ_o is no longer a basic variable [6]. Since the algebraic loop description (6) can be reformulated as the LCP(q, Q) with $Q = L^T H^{-1} L$ and $q = b - L^T v$, it can be solved using the Lemke algorithm stated in Algorithm 1. Here the solution $\hat{u}$ of the algebraic loop (2) is recovered from λ, the unique solution of the LCP using $\hat{u} = v - H^{-1} L\lambda$.

Taking advantage of the structure of Q, the LCP can be decoupled by partitioning q, s and λ as $q = [q_1^T \ q_2^T]^T$, $s = [s_1^T \ s_2^T]^T$ and $\lambda = [\lambda_1^T \ \lambda_2^T]^T$ to obtain the following sub-problems.

Case 1: $q_1 \geq 0$ and $q_2 \geq 0$. Here, the solution is trivial: $\lambda_1 = 0$, $\lambda_2 = 0$. For this case, $\hat{u} = v$ solves the algebraic loop.

Case 2: q_1 indefinite and $q_2 \geq 0$. In this case, $\lambda_2 = 0$ is a solution since $q_2 \geq 0$. The problem to be solved then reduces to

$$H^{-1}\lambda_1 + q_1 = s_1, \qquad (26a)$$

$$\lambda_1 \geq 0, \ s_1 \geq 0, \ \lambda_1^T s_1 = 0. \qquad (26b)$$

Case 3: $q_1 \geq 0$ and q_2 indefinite . Here $\lambda_1 = 0$ because $q_1 \geq 0$. This reduces the problem to

$$H^{-1}\lambda_2 + q_2 = s_2, \qquad (27a)$$

$$\lambda_2 \geq 0, \ s_2 \geq 0, \ \lambda_2^T s_2 = 0. \qquad (27b)$$

Case 4: q_1 and q_2 indefinite. No parts of the problem can be eliminated, so (24) must be completely solved.

We note that for input-constrained control applications, Case 4 is infrequent as only few constraints are violated in most cases [5]. This is based on the fact that an input u_i can only violate either the lower bound u_i^{max} or the upper bound u_i^{min} and not both at the same time. We expect that the decoupling will result in reduced computation time, especially for large problems as compared to the standard Lemke algorithm.

6 Simulation Example

We consider an example taken from [22] (also see [23]) where the plant and the corresponding unity feedback controller transfer function models are given respectively as:

$$G_{22}(s) = \frac{10}{100s+1}\begin{bmatrix} 4 & -5 \\ -3 & 4 \end{bmatrix}, K(s) = \frac{100s+1}{200s}\begin{bmatrix} 4 & 5 \\ 3 & 4 \end{bmatrix}.$$

The plant's inputs are constrained as $-1 \leq u_i \leq 1, i = 1, 2$. The unconstrained controller has been designed to track the reference input signal $w = [0.63 \ 0.79]^T$ and the performance output is selected as $z = y - w$. We simulate the system for two cases: first, anti-windup without algebraic loop and second, anti-windup with algebraic loop. For the former case, we fix $H_2 = I$ and following [22] and [20], we compute H_1 as:

$$\begin{bmatrix} -55.56 & -40.48 \\ -68.26 & -55.56 \end{bmatrix}^{[22]}, \ 10^4 \times \begin{bmatrix} -1.17 & 1.50 \\ 1.50 & -1.92 \end{bmatrix}^{[20]}.$$
$$(28)$$

We then incorporate algebraic loops following the developments in [2,27,23] for directionality compensation,[22] for static anti-windup and [20] for dynamic anti-windup. The algebraic loop feedback gains (H_2) obtained for the
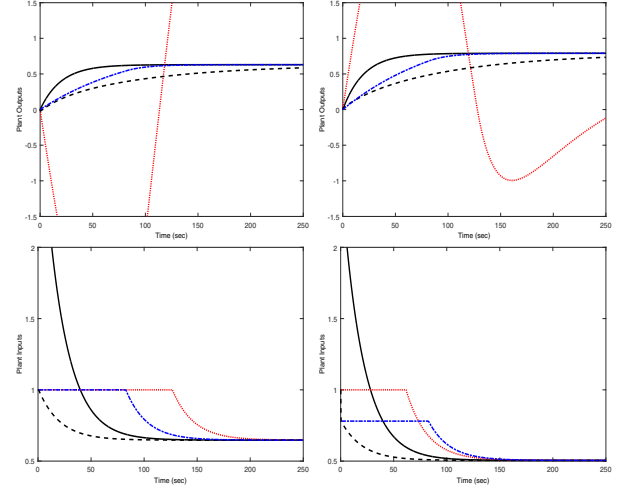

Fig. 6. Closed-loop Responses (Left-Channel 1, Right-Channel 2): Unconstrained-Solid(black), Saturated-Dotted (red), Anti-windup without algebraic loop [22]-Dashed(black), Anti-windup with algebraic loop [22]-Dot-Dashed(blue)

different configurations are given respectively as:

$$\begin{bmatrix} 0.25 & -0.32 \\ -0.32 & 0.41 \end{bmatrix}^{[27]}, \begin{bmatrix} 100.00 & -128.00 \\ -128.00 & 164.00 \end{bmatrix}^{[23]},$$

$$\begin{bmatrix} 237.15 & -303.55 \\ -303.55 & 388.92 \end{bmatrix}^{[22]} \text{ and } \begin{bmatrix} 0.0623 & -0.0798 \\ -0.0798 & 0.1022 \end{bmatrix}^{[20]}.$$

It is straightforward to check that all the computed gains are in the **P**-matrix class. Although these gains guarantee well-posedness, the closed-loop implementation of the resulting algebraic loop may be problematic. This is because the gains are ill-conditioned with condition numbers in the 10^3-10^4 range. We overcome this issue by solving the algebraic loop independently using Algorithm 1 which is well-known to handle easily such classes of problems.

For all the cases, the algorithms for solving the algebraic loops are implemented as S-function blocks in Matlab and are then embedded within the Simulink models of the system. Fig. 6 shows the input and output closed-loop responses using the anti-windup framework in Fig. 2. Here the closed-loop responses with and without algebraic loops are compared to those of the unconstrained system where control inputs are unlimited, and the uncompensated constrained system. For this framework, the static and the dynamic anti-windup result in the same responses, so we show plots only for the static anti-windup. Fig. 7 shows the closed-loop responses using the anti-windup framework in Fig. 3 with and without algebraic loop as compared to the unconstrained closed-loop response. Here, the simulation corresponding to the anti-windup without algebraic loop requires very high sampling, resulting in a ringing effect on the
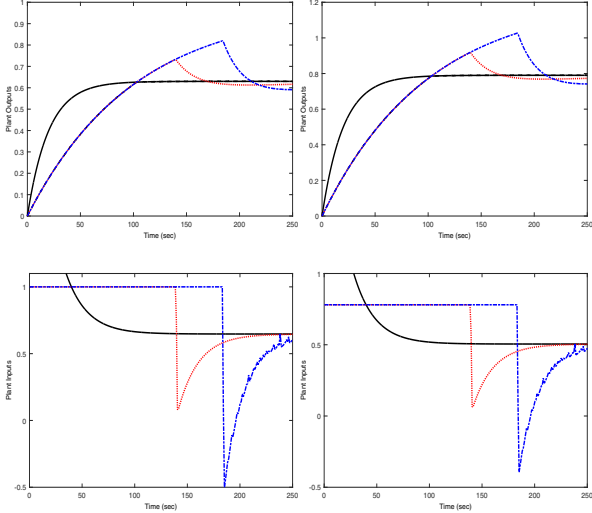
8

Fig. 7. Closed-loop Input Responses (Left-Channel 1, Right-Channel 2): Unconstrained-Solid (black), Anti-windup without algebraic loop-(not shown), Anti-windup with algebraic loop[27]-Dot-Dashed(blue) and Anti-windup with algebraic loop [20]-Dotted (red)

input response; hence it is not shown. It is well known in the literature that this scheme tends to place the compensator poles far to the left of the s-plane [30]. The computed H_1 matrices corresponding to [27] and [20] are respectively given as:

$$\begin{bmatrix} -3.27 & 0.00 \\ 0.00 & -3.27 \end{bmatrix}^{[27]} \quad \text{and} \quad \begin{bmatrix} -10.15 & -0.10 \\ -0.10 & -10.10 \end{bmatrix}^{[20]}. \quad (29)$$

Comparing the H_1 in (29) for anti-windup with algebraic loops to those in (28) for anti-windup without algebraic loop shows the conditioning effect of algebraic loops on the anti-windup design.

These simulation results demonstrate the benefits that accrue from incorporating algebraic loops into linear anti-windup design. While no improved performance over existing methods is guaranteed (e.g. [28,12,20]), the new framework is general and its implementation is particularly efficient in the LCP framework.

Acknowledgements

7 Conclusion

We have presented a unified framework for addressing multivariable algebraic loops both in terms of well-posedness and practical implementation in linear anti-windup control. Several existing well-posedness results appear as special cases of the unified framework.

A   Appendix: Proofs of Technical Results

Lemma 1: It is straightforward that any solution $\hat{u}_i : i = 1, \cdots, m$ of the decentralized saturation defined by (3) must satisfy:

$$(\hat{u}_i - u_i) \begin{cases} < 0; & \hat{u} = u_i^{max}, \\ = 0; & u_i^{min} \le \hat{u}_i \le u_i^{max}, \\ > 0; & \hat{u}_i = u_i^{min}. \end{cases} \quad (A.1)$$

Using KKT formalism (e.g.[10,24]), there exist Lagrangian multipliers λ_i^1, λ_i^2; $i = 1, \cdots, m$ such that the following (necessary and sufficient) optimality conditions hold:

$$\begin{aligned} \hat{u}_i - u_i + \lambda_i^1 - \lambda_i^2 = 0; & i = 1, \cdots, m, \\ \hat{u}_i - u_i^{max} \le 0; \ -\hat{u}_i + u_i^{min} \le 0; & i = 1, \cdots, m, \\ \lambda_i^1 \ge 0; \ \lambda_i^1(\hat{u}_i - u_i^{max}) = 0; & i = 1, \cdots, m, \\ \lambda_i^2 \ge 0; \ -\lambda_i^2(\hat{u}_i - u_i^{min}) = 0; & i = 1, \cdots, m. \end{aligned} \quad (A.2)$$

Using L and b defined in (5) and $\lambda = [(\lambda^1)^T \ (\lambda^2)^T]^T$, the KKT conditions in (A.2) can be expressed in vector form as

$$\begin{aligned} \hat{u} - u + L\lambda &= 0, \\ L^T\hat{u} - b &\le 0, \quad (A.3) \\ \lambda \ge 0; \ \lambda^T(L^T\hat{u} - b) &= 0. \end{aligned}$$

Substituting $u = Hv - (H - I)\hat{u}$ into (A.3) gives (6).

Lemma 4: **Part 1 [Feasibility of solution]:** Since H is assumed to be a **P**-matrix (and hence nonsingular), H^{-1} exists and a **P**-matrix (see [16]). For our case, it is easy to construct a λ such that (8a) holds. Choose $\lambda^1 \in \mathbb{R}^m$ such that $\lambda^1 \ge 0$ and $\lambda^1 - Hv \ge 0$. Then choose $\lambda = [(\lambda^1)^T \ (\lambda^1 - Hv)^T]^T$. By construction $\lambda \ge 0$. Furthermore, $L\lambda = Hv$ so

$$q + Q\lambda = b - L^Tv + L^TH^{-1}L\lambda = b \ge 0. \quad (A.4)$$

Hence λ is a feasible solution of the LCP(q, Q).

Part 2 [Uniqueness of solution]: Assume λ^* solves the LCP(q, Q). Then λ^* and $\gamma^* = q + Q\lambda^*$ satisfy (8). Let λ be another solution of the LCP(q, Q) such that λ and $\gamma = q + Q\lambda$ also satisfy (8). Then we have

$$(\lambda^* - \lambda)^T(\gamma^* - \gamma) = (\lambda^* - \lambda)^TQ(\lambda^* - \lambda). \quad (A.5)$$

By enforcing (8), we have that for all $i = 1, \cdots, 2m$

$$0 \ge (\lambda^* - \lambda)_i(\gamma^* - \gamma)_i = (\lambda^* - \lambda)_i[Q(\lambda^* - \lambda)]_i. \quad (A.6)$$

Since H is a **P**-matrix and $Q = L^T H^{-1} L$, it follows that Q is column sufficient i.e. the following implication holds

$$[d_i(Qd)_i \leq 0 \ \forall \ i] \Longrightarrow [d_i(Qd)_i = 0 \ \forall \ i]. \qquad (A.7)$$

See [6, Chapter 3.5] and [1, Lemma 2]. Thus by the column sufficiency of Q, we have $Q(\lambda^* - \lambda) = 0$ and hence the uniqueness of γ^*.

To show the uniqueness of λ^* when $b > 0$, observe that λ^* and λ are both solutions. The left part of (A.6) implies

$$\lambda_i^* \gamma_i = \lambda_i \gamma_i^* = 0 \text{ for } i = 1, \cdots, m. \qquad (A.8)$$

Suppose we write $b, \lambda^*, \lambda \in \mathbb{R}^{2m}$ as $b = [(b^1)^T, (b^2)^T]^T$, $\lambda^* = [(\lambda^{*1})^T, (\lambda^{*2})^T]^T$ and $\lambda = [(\lambda^{*1}+\alpha)^T, (\lambda^{*2}+\alpha)^T]^T$ for some $\alpha \in \mathbb{R}^m$. This, together with (A.8) leads to

$(\lambda^{*1} + \alpha)_i[b^1 - v + H^{-1}(\lambda^{*1} - \lambda^{*2})]_i = 0$
$(\lambda^{*2} + \alpha)_i[b^2 + v - H^{-1}(\lambda^{*1} - \lambda^{*2})]_i = 0$
$\lambda_i^{*1}[b^1 - v + H^{-1}(\lambda^{*1} - \lambda^{*2})]_i = 0$ and
$\lambda_i^{*2}[b^2 + v - H^{-1}(\lambda^{*1} - \lambda^{*2})]_i = 0$ for $i = 1, \cdots, m$.

After simple algebraic manipulation, we have

$$\alpha_i[b^1 + b^2]_i = 0 \text{ for } i = 1, \cdots, m. \qquad (A.9)$$

If $b^1 > 0$ and $b^2 > 0$ then (A.9) requires $\alpha = 0$ and hence the uniqueness of λ^*.

Proposition 5; Part 1: Since H is assumed to be a **P**-matrix (and hence nonsingular), we can eliminate (6a) from (6) by rewriting it as

$$\hat{u} = v - H^{-1} L\lambda. \qquad (A.10)$$

Substituting (A.10) into (6b) gives

$$L^T H^{-1} L\lambda + b - L^T v \geq 0, \qquad (A.11a)$$
$$\lambda \geq 0; \ \lambda^T(L^T H^{-1} L\lambda + b - L^T v) = 0. \qquad (A.11b)$$

Defining $Q = L^T H^{-1} L$ and $q = b - L^T v$ in (A.11) gives the LCP(q, Q). Let λ and λ^* be any two solutions of the LCP(q, Q). From Lemma 4 and the column sufficiency of Q, we have

$$(L(\lambda^* - \lambda))_i[H^{-1}L(\lambda^* - \lambda)]_i = 0 \text{ for } i = 1, \cdots, 2m.$$

Since H^{-1} is a **P**-matrix, we must have $L(\lambda^* - \lambda) = 0$ and hence the uniqueness of $L\lambda$. It then follows from (A.10) that $\hat{u}$ is the unique solution of the algebraic loop for any v. Well-posedness of the algebraic loop follows.

Proposition 5; Part 2: Let $\hat{u}^i$ be the unique solution of the algebraic loop (2) corresponding to v^i for $i = 1, 2$. If

we introduce a slack variable s, (6) can be expressed as:

$$H\hat{u} - Hv + L\lambda = 0, \qquad (A.12a)$$
$$s = b - L^T \hat{u}, \qquad (A.12b)$$
$$s \geq 0, \lambda \geq 0; \ \lambda^T s = 0. \qquad (A.12c)$$

From (A.12b), we have

$$(\lambda^1 - \lambda^2)^T(s^1 - s^2) = -(\lambda^1 - \lambda^2)^T L^T(\hat{u}^1 - \hat{u}^2). \ (A.13)$$

Using (A.12a) in (A.13) yields

$$(\lambda^1 - \lambda^2)^T(s^1 - s^2) = -(v^1 - v^2)^T H^T(\hat{u}^1 - \hat{u}^2)$$
$$+ (\hat{u}^1 - \hat{u}^2)^T H^T(\hat{u}^1 - \hat{u}^2).$$

Applying (A.12c) and re-arranging gives

$$(\hat{u}^1 - \hat{u}^2)^T H(v^1 - v^2) \geq (\hat{u}^1 - \hat{u}^2)^T H(\hat{u}^1 - \hat{u}^2). \ (A.14)$$

Since H is a **P**-matrix, the following inequality holds (see e.g. [6, pg 479])

$$\max_i \hat{u}_i^T[H\hat{u}]_i \geq c||\hat{u}||^2, \text{ for some } c > 0. \qquad (A.15)$$

Combining (A.14) and (A.15), we deduce that

$$(\hat{u}^1 - \hat{u}^2)^T H(v^1 - v^2) \geq c||\hat{u}^1 - \hat{u}^2||^2, \qquad (A.16)$$

from which the Lipschitz property of $\hat{u}$ as a function v follows.

Proposition 7: We only need to show that any matrix $H \in \mathbb{R}^{m \times m}$ satisfying condition (10) or (11) is a **P**-matrix. The conclusion then follows from Proposition 5. We proceed as follows:

Part 1: Let X be a positive-diagonal matrix such that (10) holds. Since $HX + XH^T$ is the symmetric part of HX, and since a matrix is positive definite if and only if its symmetric part is positive definite (e.g. see [17, Remark 1]), it follows that H is diagonally positive stable and so are all its the principal submatrices. From Lyapunov theorem (e.g. see [7, Theorem 1]), all the eigenvalues of H and its principal submatrices have positive real parts. As H is real, any complex eigenvalues appear in conjugate pairs and thus all the principal minors of H must be positive. We conclude that H is a **P**-matrix.

Part 2: Now let X be a positive-diagonal matrix such that (11) holds. This condition implies that HX is strictly diagonally dominant with positive diagonal entries. It follows from the Gersgorin theorem [14, Theorem 6.1.10] that all the eigenvalues of HX have positive real part. This implies that H and all its principal submatrices are diagonally positive stable. Following the

same argument as for Part 1, we conclude that H is a **P**-matrix.

Corollary 11: Since a positive definite matrix is a **P**-matrix, it follows from Proposition 5 that the mLCP (6) has (v^*, λ^*) as its unique solution if H is symmetric positive definite. On the other hand, mLCP (6) describes the necessary optimality conditions for $\hat{u}^*$ to be a solution of QP (20)[10]. Since H is positive definite, the objective function of QP (20) is strictly convex. Hence, the KKT optimality conditions are also sufficient and $\hat{u}^*$ must be global and unique.

References

[1] A.A. Adegbege and W.P. Heath. Multivariable algebraic loops with complementarity constraints enforcing some KKT conditions. In *52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2014*, pages 1033–1039, Monticello, Illinois, Sept 2014.

[2] A.A. Adegbege and W.P. Heath. Directionality compensation for linear multivariable anti-windup synthesis. *International Journal of Control*, 88(11):2392–2402, 2015.

[3] A.A. Adegbege and W.P. Heath. Multivariable algebraic loops in linear anti-windup implementations. In *Proceedings of the 23rd Mediterranean Conference on Control and Automation*, pages 514–519, Torremolinos, Spain, June 2015.

[4] A.A. Adegbege and W.P. Heath. Robust synthesis for directionality compensation. *IET Control Theory & Application*, 10(3):312–319, Feb 2016.

[5] E.F. Camacho. Constrained generalized predictive control. *IEEE Transactions on Automatic Control*, 38(2):327–332, Feb 1993.

[6] R.W. Cottle, J. Pang, and R.E. Stone. *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2009.

[7] G.W. Cross. Three types of matrix stability. *Linear Algebra and its Applications*, 20(3):253 – 263, 1978.

[8] F.J. D'Amato, M.A. Rotea, A.V. Megretski, and U.T. Jönsson. New results for analysis of systems with repeated nonlinearities. *Automatica*, 37(5):739 – 747, 2001.

[9] C.A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, Inc., Orlando, FL, USA, 1975.

[10] R. Fletcher. *Practical Methods of Optimization*. John Willey & Sons, Chichester, 1987.

[11] G. Grimm, J. Hatfiled, A.R. Teel, M.C. Turner, and L. Zaccarian. Anti-windup for stable linear systems with input saturation: an LMI based synthesis. *IEEE Transactions on Automatic Control*, 48(9):1509–1525, 2003.

[12] G. Grimm, A.R. Teel, and L. Zaccarian. Establishing the lipschitz properties of multivariable algebraic loops with incremental sector nonlinearities. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui Hawaii, December 2003.

[13] W.P. Heath and G. Li. Multipliers for model predictive control with structured input constraints. *Automatica*, 46(3):562–568, 2010.

[14] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[15] T. Hu, A.R. Teel, and L. Zaccarian. Stability and performance for saturated systems via quadratic and nonquadratic lyapunov functions. *IEEE Transactions on Automatic Control*, 51(11):1770–1786, Nov 2006.

[16] Charles R Johnson and Michael J Tsatsomeros. Convex sets of nonsingular and p:–matrices. *Linear and Multilinear Algebra*, 38(3):233–239, 1995.

[17] C.R. Johnson. Positive definite matrices. *The American Mathematical Monthly*, 77(3):259 – 264, 1970.

[18] M.V. Kothare, P.J. Campo, M. Morari, and C.N. Nett. A unified framework for the study of anti-windup designs. *Automatica*, 30(12):1869–1883, 1994.

[19] R.M. Levenson and A.A. Adegbege. Analog circuit for real-time optimization of constrained control. In *2016 American Control Conference (ACC)*, pages 6947–6952, July 2016.

[20] G. Li, G. Herrmann, D.P. Stoten, J. Tu, and M.C. Turner. A novel robust disturbance rejection anti-windup framework. *International Journal of Control*, 84(1):123–137, 2011.

[21] J.L. Morales, J. Nocedal, and M. Smelyanskiy. An algorithm for the fast solution of symmetric linear complementarity problems. *Numerische Mathematik*, 111(2):251–266, 2008.

[22] E.F. Mulder, M.V. Kothare, and M. Morari. Multivariable anti-windup controller synthesis using linear matrix inequalities. *Automatica*, 37:1407–1416, 2001.

[23] Y. Peng, D. Vrančić, R Hanus, and S. R. Weller. Anti-windup designs for multivariable controllers. *Automatica*, 34(12):1559–1565, 1998.

[24] J.A. Primbs and M. Giannelli. Kuhn -Tucker-based stability conditions for systems with saturation. *IEEE Transactions on Automatic Control*, 46(10):1643–1647, 2001.

[25] S. Sajjadi-Kia and F. Jabbari. Modified anti-windup compensators for stable plants. *IEEE Transactions on Automatic Control*, 54(8):1934–1939, Aug 2009.

[26] M. Soroush and K.R. Muske. Analytical model predictive control. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 163–179. Birkhäuser Verlag Bassel, Switzerland, 2000.

[27] M. Soroush and S. Valluri. Optimal directionality compensation in processes with input saturation non-linearities. *International Journal of Control*, 72(17):1555–1564, 1999.

[28] A. Syaichu-Rohman and R.H. Middleton. On the robustness of multivariable algebraic loops with sector nonlinearities. In *Proceedings of the 41st IEEE Conference on Control and Decision*, pages 1054–1059, Las Vegas, Nevada, December 2002.

[29] A. Syaichu-Rohman, R.H. Middleton, and M.M. Seron. A multivariable nonlinear algebraic loop as a QP with application to MPC. In *Proceedings of the European Control Conference*, pages 6193–6198, Cambridge, September 1-4 2003.

[30] M.C. Turner, G. Herrmann, and I. Postlethwaite. Incorporating robustness requirements into antiwindup design. *IEEE Transactions on Automatic Control*, 52(10):1842–1855, 2007.

[31] J.C. Willems. *The Analysis of Feedback Systems*. MIT Press, Cambridge, Massacchusetts, 1971.

[32] S.J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1997.

[33] L. Zaccarian and A.R. Teel. *Modern Anti-windup Synthesis*. Princeton Press, New Jersey, 2011.