

Anti-windup Compensation for Nonlinear Systems via Gradient Projection: Application to Adaptive Control

Justin Teo and Jonathan P. How

Abstract—Control saturation is an important limitation in practical control systems and it is well known that performance degradation or instability may result if this limitation is not effectively addressed. Using ideas from the gradient projection method in nonlinear programming, we propose a new anti-windup scheme for multi-input, multi-output nonlinear dynamic controllers. The key idea is to project the controller state update law onto the tangent plane of the active saturation constraints. To do this, we first extend the gradient projection method to the continuous-time case that can accommodate *multiple* nonlinear constraints. This is then used for anti-windup compensation, resulting in a hybrid controller that switches its state update law over arbitrary combinations of saturating controls. Simulations on a nonlinear two-link robot driven by an adaptive sliding mode controller illustrates its effectiveness and limitations.

I. INTRODUCTION

Control saturation is inherent in virtually all practical control systems and presents a significant limitation to achievable closed loop performance. It is well known that when this limitation is not effectively accounted for, the closed loop system may suffer from severe performance degradation, or even destabilize [1], [2], [3]. This phenomenon and its compensation is termed *windup* and *anti-windup compensation* respectively [2]. It was recognized in the recent survey paper [3] that anti-windup compensation for nonlinear systems remains largely an *open problem*. To this end, [4] and relevant references in [3] represent some recent advances. These anti-windup schemes are applicable to feedback linearizable nonlinear systems, or Euler-Lagrange systems [4]. In contrast, the anti-windup scheme to be presented, called *Gradient Projection Anti-windup* (GPAW), is applicable to a much larger class of *nonlinear* systems and controllers.

One key element of GPAW is the use of gradient projection, which is widely used to solve constrained nonlinear programs [5], [6]. In the absence of any active constraints, it reduces to the steepest descent method [5], [6]. As shown in [7, Appendix B], by taking the limit as the stepsize of the gradient projection method is decreased to zero, one obtains the continuous time gradient projection method. In the context of adaptive control, this has been successfully applied to bound parameter estimates (or equivalently, the

state of the adaptive controller) in some a priori known region in the parameter space [7, Sections 4.4, 8.4.2, and 8.5.5]. We first extend the gradient projection method of [5], [6] to the continuous-time case that can accommodate *multiple* nonlinear constraints. For each controller output, there corresponds two saturation constraints. Using the generalized continuous-time gradient projection method, we project the controller state update law onto the intersection of all tangent planes of all active saturation constraints to construct the GPAW compensated controller. In line with the classical interpretation that windup arises due to inconsistencies between the controller state and output [1], the GPAW compensated controller maintains such consistency in an approximate sense in the presence of arbitrary combinations of saturating controls.

The GPAW scheme, while starkly different, has some similarities to many of the existing anti-windup schemes in the following sense: (i) it can be viewed as a generalization of the “stop integration” heuristic as outlined in [8], to the case of nonlinear multi-input, multi-output (MIMO) controllers; (ii) it has a certain optimization flavor reminiscent of the numerous Linear Matrix Inequalities based approaches to anti-windup compensation surveyed in [3]. As will be shown, the GPAW scheme requires the solution to a *combinatorial* optimization sub-problem; (iii) similar to many of the classical anti-windup schemes [1], it attempts to minimize the inconsistency between the controller state and outputs that arise under control saturation.

This first paper strives to give insights and intuition of the GPAW scheme and demonstrate its viability. Future papers will then investigate key properties of the approach. The paper is organized as follows. The problem statement is presented in Section II. To illustrate the basic ideas of GPAW, decoupled controllers are considered in Section III. The key ideas of gradient projection are recalled in Section IV, and Section V extends the gradient projection method of [5], [6] to the continuous-time case. The main theme of this paper, which is the GPAW scheme, is presented in Section VI. Section VII illustrates the effectiveness and limitations of the GPAW scheme by applying it to a nonlinear two-link robot controlled by an adaptive sliding mode controller [9]. Finally, we highlight some limitations of the GPAW scheme in Section VIII.

The following conventions will be used in the sequel. For a scalar function $f: (\mathbf{x}, \mathbf{u}) \mapsto f(\mathbf{x}, \mathbf{u})$, its partial derivative $\frac{\partial f}{\partial \mathbf{x}}$ is a *row* vector, and its gradient $\nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T$ is a *column* vector. For inequalities involving vectors, the inequality is to be interpreted element-wise. Let A and B be

J. Teo is a graduate student, MIT Department of Aeronautics and Astronautics. csteo@mit.edu

J. How is Professor, MIT Department of Aeronautics and Astronautics, Senior Member IEEE. jhow@mit.edu

The first author acknowledges Prof. Jean-Jacques Slotine and Dr. Han-Lim Choi for their valuable insights, and DSO National Laboratories, Singapore, for financial support. Reviewers’ comments are very much appreciated. Research funded in part by AFOSR grant FA9550-08-1-0086.

logical statements that evaluate to *true* or *false*. Then $\neg A$, $A \wedge B$ and $A \vee B$ correspond to the statements “NOT A ”, “ A AND B ”, and “ A OR B ” respectively. Let \mathcal{I} and \mathcal{J} be two sets. The cardinality of \mathcal{I} will be denoted by $|\mathcal{I}|$, and $\mathcal{I} \setminus \mathcal{J}$ is the relative complement of \mathcal{J} in \mathcal{I} .

II. PROBLEM STATEMENT

For some $u_{\min}, u_{\max} \in \mathbb{R}$ satisfying $u_{\min} < u_{\max}$, the *scalar* saturation function is defined by

$$\text{sat}(u, u_{\min}, u_{\max}) = \begin{cases} u_{\max}, & \text{if } u_{\max} < u, \\ u, & \text{if } u_{\min} \leq u \leq u_{\max}, \\ u_{\min}, & \text{if } u < u_{\min}. \end{cases}$$

Let $u_{i\min} < u_{i\max}$ for all $i \in \{1, 2, \dots, m\}$. For any $\mathbf{u} = [u_1, u_2, \dots, u_m]^T \in \mathbb{R}^m$, the vector saturation function is defined by

$$\text{sat}(\mathbf{u}) = [\text{sat}(u_1, u_{1\min}, u_{1\max}), \dots, \text{sat}(u_m, u_{m\min}, u_{m\max})]^T \in \mathbb{R}^m,$$

where its dependence on $u_{i\min}$ and $u_{i\max}$ has been dropped.

Consider the *input constrained* system described by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \text{sat}(\mathbf{u})), & \mathbf{x}(0) &= \mathbf{x}_0, \\ \mathbf{y} &= \mathbf{g}(\mathbf{x}, \text{sat}(\mathbf{u})), \end{aligned} \quad (1)$$

and the *nominal* controller described by

$$\begin{aligned} \dot{\mathbf{x}}_c &= \mathbf{f}_c(\mathbf{x}_c, \mathbf{y}, \mathbf{r}(t)), & \mathbf{x}_c(0) &= \mathbf{x}_{c0}, \\ \mathbf{u}_c &= \mathbf{g}_c(\mathbf{x}_c, \mathbf{y}, \mathbf{r}(t)), \end{aligned} \quad (2)$$

where $\mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, are the state, initial state and control input of system (1), $\mathbf{y} \in \mathbb{R}^p$ is the output of system (1) and measurement for the controller (2), $\mathbf{x}_c, \mathbf{x}_{c0} \in \mathbb{R}^q$, $\mathbf{r}(t) \in \mathbb{R}^r$, $\mathbf{u}_c \in \mathbb{R}^m$, are the state, initial state, *instantaneous* exogenous input, and output of the controller (2). Let \mathcal{R} be a class of signals evolving in \mathbb{R}^r , eg. $C([0, \infty), \mathbb{R}^r)$, the vector space of continuous functions $[0, \infty) \mapsto \mathbb{R}^r$. Assume that the functions $\mathbf{f}: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, $\mathbf{g}: \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^m$, $\mathbf{f}_c: \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^r \rightarrow \mathbb{R}^q$, and $\mathbf{g}_c: \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^r \rightarrow \mathbb{R}^m$, are such that the *uncompensated* closed loop system defined by (1), (2) and

$$\mathbf{u} = \mathbf{u}_c,$$

is well-posed, ie. existence and uniqueness of solutions are assured for all $t \geq 0$, $(\mathbf{r}, \mathbf{x}_0, \mathbf{x}_{c0}) \in \mathcal{R} \times \mathbb{R}^n \times \mathbb{R}^q$, and achieves at least some *local* stability properties. Assume further that the nominal controller (2) is known exactly.

Let the *anti-windup compensated* controller to be designed be described by

$$\begin{aligned} \dot{\mathbf{x}}_{\text{aw}} &= \mathbf{f}_{\text{aw}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}(t)), & \mathbf{x}_{\text{aw}}(0) &= \mathbf{x}_{\text{aw}0}, \\ \mathbf{u}_{\text{aw}} &= \mathbf{g}_{\text{aw}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}(t)), \end{aligned} \quad (3)$$

where $\mathbf{x}_{\text{aw}} \in \mathbb{R}^{\bar{q}}$ (possibly with $\bar{q} = q$), taking the same inputs $(\mathbf{y}, \mathbf{r}(t))$ and having an m dimensional output $\mathbf{u}_{\text{aw}} \in \mathbb{R}^m$. The *anti-windup compensated* closed loop system is defined by (1), (3) and

$$\mathbf{u} = \mathbf{u}_{\text{aw}}.$$

The problem is to design $\mathbf{f}_{\text{aw}}: \mathbb{R}^{\bar{q}} \times \mathbb{R}^p \times \mathbb{R}^r \rightarrow \mathbb{R}^{\bar{q}}$ and $\mathbf{g}_{\text{aw}}: \mathbb{R}^{\bar{q}} \times \mathbb{R}^p \times \mathbb{R}^r \rightarrow \mathbb{R}^m$, and determine an initialization $\mathbf{x}_{\text{aw}0} \in \mathbb{R}^{\bar{q}}$ such that:

- (i) for every $(\mathbf{r}, \mathbf{x}_0, \mathbf{x}_{c0}) \in \mathcal{R} \times \mathbb{R}^n \times \mathbb{R}^q$ such that the controls never saturate for the uncompensated system, ie. $\text{sat}(\mathbf{u}_c) \equiv \mathbf{u}_c$, the control signal of the anti-windup compensated system satisfy $\mathbf{u}_{\text{aw}} \equiv \mathbf{u}_c$. In other words, nominal system performance is recovered whenever no controls saturate.
- (ii) when $(\mathbf{r}, \mathbf{x}_0, \mathbf{x}_{c0})$ are such that some controls saturate for the uncompensated nominal system, ie. $\text{sat}(\mathbf{u}_c(t)) \neq \mathbf{u}_c(t)$, for some non-trivial interval of time, then the performance of the anti-windup compensated system degrades gracefully.

By graceful performance degradation, we appeal to the intuitive notion that as the saturation conditions becomes more severe, the system performance degrade more, while preserving some stability properties.

III. DECOUPLED NONLINEAR MIMO CONTROLLERS

This section illustrates the intuition behind the GPAW scheme for decoupled nonlinear MIMO controllers before considering the general case. In the classical anti-windup literature [1], windup is interpreted as arising from an inconsistency between the controller state and outputs when some controls saturate. One of the earliest ad-hoc schemes for anti-windup compensation is to stop the integration of the controller state whenever the control saturates [8]. This maintains consistency between the controller state and output in an approximate sense.

Extending this simple idea for a *decoupled* nonlinear MIMO controller is straightforward. For example, consider a *decoupled* two-state, two-output controller

$$\begin{aligned} \dot{x}_{c1} &= f_{c1}(x_{c1}, y_1, r_1), & \dot{x}_{c2} &= f_{c2}(x_{c2}, y_2, r_2), \\ u_{c1} &= g_{c1}(x_{c1}, y_1, r_1), & u_{c2} &= g_{c2}(x_{c2}, y_2, r_2). \end{aligned}$$

In this case, x_{ci} is the state for the control output u_{ci} for $i \in \{1, 2\}$, and u_{ci} is independent of (y_j, r_j) for $j \neq i$. Therefore, the “stop integration” rule is simply to set $\dot{x}_{ci} = 0$ whenever $\text{sat}(u_{ci}, u_{i\min}, u_{i\max}) \neq u_{ci}$.

This scheme can be enhanced by stopping integration only when the associated control is saturated and integration in the nominal direction leads to worse violation of the saturation constraint. More concretely, the controller state update law with this modification can be written as

$$\dot{x}_{ci} = \begin{cases} 0, & \text{if } A_i, \\ f_{ci}(x_{ci}, y_i, r_i), & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} A_i &= \left((u_{ci} \geq u_{i\max}) \wedge \left(\frac{\partial g_{ci}}{\partial x_{ci}} f_{ci}(x_{ci}, y_i, r_i) > 0 \right) \right) \\ &\quad \vee \left((u_{ci} \leq u_{i\min}) \wedge \left(\frac{\partial g_{ci}}{\partial x_{ci}} f_{ci}(x_{ci}, y_i, r_i) < 0 \right) \right). \end{aligned}$$

The idea behind this last modification is well known for Proportional-Integral-Derivative (PID) type controllers [10, Conditionally Freeze Integrator method].

IV. GRADIENT PROJECTION

Section IV motivated ideas of GPAW using the simplified problem of decoupled controllers. Now, consider the extension of these same ideas for general nonlinear *coupled* MIMO controllers. In this case, it is immediately clear that we cannot selectively stop the integration of any particular element of the controller state vector, since each element will in general affect more than one controller output. Doing so may adversely affect those controller outputs that have not yet reached saturation. Furthermore, just stopping integration on all states when any controller output is saturated is far too conservative. What is needed then, at a fixed point in time, is a way to *update the controller state vector in the nominal direction as much as possible, while attempting not to aggravate any existing saturation conditions*.

One way to achieve this is by gradient projection [5], [6]. The gradient projection method solves constrained nonlinear programming problems of the form

$$\min_{\mathbf{x} \in \mathbb{R}^q} J(\mathbf{x}), \quad \text{subject to } \mathbf{h}(\mathbf{x}) \leq \mathbf{0}, \quad (4)$$

where $\mathbf{x} \in \mathbb{R}^q$, $J(\mathbf{x})$ is a possibly nonlinear scalar function, and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_k(\mathbf{x})]^T$ is a set of k possibly nonlinear functions. In its basic form, it is a very powerful method, with only very mild differentiability requirements on the functions $J(\mathbf{x})$ and $h_i(\mathbf{x})$, $i \in \{1, 2, \dots, k\}$. However, some additional assumptions like convexity, boundedness of feasible region etc., must be imposed to ensure convergence to the global minimum [5], [6]. Note that only the underlying idea of gradient projection is relevant to the GPAW scheme.

In the absence of any constraints, the gradient projection method reduces to the steepest descent method [5], [6]. The key mechanism that enables the method to maintain feasibility is gradient projection. Each of the k inequalities $h_i(\mathbf{x}) \leq 0$, $i \in \{1, 2, \dots, k\}$, defines a hypersurface G_i in \mathbb{R}^q that forms the boundary of the feasible region $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^q \mid \mathbf{h}(\mathbf{x}) \leq \mathbf{0}\}$. On each point $\tilde{\mathbf{x}}$ of the boundary of \mathcal{S} , each hypersurface G_i that contains $\tilde{\mathbf{x}}$ has an associated *supporting hyperplane* $H_i(\tilde{\mathbf{x}})$ that is tangent to G_i at $\tilde{\mathbf{x}}$. The normal of H_i at the point $\tilde{\mathbf{x}}$ is the gradient of $h_i(\mathbf{x})$ at $\tilde{\mathbf{x}}$, which will point “away” from \mathcal{S} . These are illustrated in Fig. 1.

Similar to many optimization methods, the gradient projection method generates a sequence $\{\mathbf{x}_n\}$, the limiting point of which would be the solution to the nonlinear program (4). Consider now, the case where all $h_i(\mathbf{x})$ are affine functions of \mathbf{x} [5]. Then G_i coincides with H_i , and the boundary of \mathcal{S} are all hyperplanes. At a particular point \mathbf{x}_n that lies in the interior of \mathcal{S} (cf. \mathbf{x}_0 in Fig. 1), the basic step is taken in a direction to decrease $J(\mathbf{x}_n)$, ie. in the negative gradient direction, $-\nabla J(\mathbf{x}_n)$, much like the steepest descent method. When \mathbf{x}_n lies on the boundary of \mathcal{S} (cf. \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 in Fig. 1), the step is taken in a direction “closest” to $-\nabla J(\mathbf{x}_n)$ while remaining within \mathcal{S} . In this case, if $-\nabla J(\mathbf{x}_n)$ points into the interior of \mathcal{S} , the nominal direction is taken. Otherwise, $-\nabla J(\mathbf{x}_n)$ is projected onto the intersection of the *smallest set of linearly independent* hyperplanes H_i that

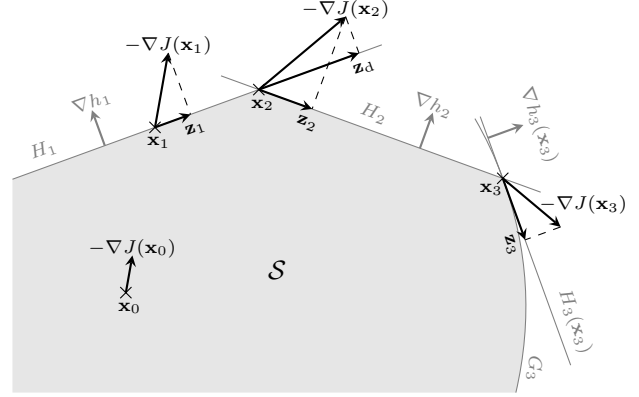


Fig. 1. Gradient Projection Method. \mathcal{S} is the feasible region, bounded by the hypersurfaces H_1 , H_2 and G_3 . The supporting hyperplane of G_3 at \mathbf{x}_3 is $H_3(\mathbf{x}_3)$. The projection of $-\nabla J(\mathbf{x}_i)$ onto H_i yields \mathbf{z}_i , while \mathbf{z}_d is the projection of $-\nabla J(\mathbf{x}_2)$ onto H_1 . Notice that to maintain feasibility at \mathbf{x}_2 , it is sufficient to project onto H_2 . In contrast, projection onto the intersection of *both* active constraints corresponding to H_1 and H_2 will yield the zero vector.

corresponds to active constraints ($h_i(\mathbf{x}_n) \geq 0$) that can keep \mathbf{x}_{n+1} within \mathcal{S} (cf. \mathbf{z}_1 , \mathbf{z}_2 and \mathbf{z}_3 in Fig. 1). The step is then taken in this new direction, and if some active constraints are nonlinear (cf. G_3 in Fig. 1), a correction is added to drive the new point \mathbf{x}_{n+1} back to \mathcal{S} .

It is important to note that this *smallest* set of hyperplanes may exclude some active constraints, but nonetheless ensures that such exclusion will not cause further constraint violations. This case is illustrated in Fig. 1, at the point \mathbf{x}_2 . Notice here that taking a step in the direction of $-\nabla J(\mathbf{x}_2)$ will violate both constraints corresponding to H_1 and H_2 . If we project onto *both* of these active constraints, (in other words, the intersection of H_1 and H_2), the result is the zero vector, and no progress can be made. However, projecting onto H_2 alone, both constraints will be satisfied, and progress can be made in the direction \mathbf{z}_2 . Notice that projecting onto H_1 to get \mathbf{z}_d is ineffective, since taking a step in this direction will violate the constraint corresponding to H_2 .

The preceding summarizes the pertinent features of the gradient projection method relevant to the GPAW scheme, but [5], [6] provide more details.

V. GENERALIZED CONTINUOUS-TIME GRADIENT PROJECTION METHOD

As shown in [7, Appendix B], the continuous-time gradient projection method can be obtained by taking the limit as the stepsize of the original gradient projection method [5], [6] is decreased to zero. This has been used successfully in the context of adaptive control to bound parameter estimates in some a priori known region in the parameter space [7, Sections 4.4, 8.4.2, and 8.5.5]. Ref. [11] presents another popular projection scheme used in adaptive control. While each has its merits, both of these methods are limited to projection with respect to a single constraint. This section presents a generalized continuous-time gradient projection method that can accommodate *multiple* nonlinear constraints.

Let $\mathcal{I}_n = \{1, 2, \dots, n\}$ where n is some positive integer. Then \mathcal{I}_k is the set of all indices corresponding to the k

constraints in problem (4). Let $\mathcal{I} \subset \mathcal{I}_k$ be some index set of cardinality $|\mathcal{I}| = s, s \leq k$. For $s > 0$, ie. $\mathcal{I} \neq \emptyset$, let $\sigma_{\mathcal{I}}: \mathcal{I}_s \rightarrow \mathcal{I}$ be a (non-unique) bijective map that assigns an integer in \mathcal{I} to each integer in \mathcal{I}_s . For $s > 0$, define the $q \times s$ matrix

$$\mathbf{N}_{\mathcal{I}}(\mathbf{x}) = [\nabla h_{\sigma_{\mathcal{I}}(1)}(\mathbf{x}), \nabla h_{\sigma_{\mathcal{I}}(2)}(\mathbf{x}), \dots, \nabla h_{\sigma_{\mathcal{I}}(s)}(\mathbf{x})].$$

It can be seen that $\mathbf{N}_{\mathcal{I}}(\mathbf{x})$ is the concatenation of those gradient vectors whose indices are in \mathcal{I} in some order determined by the map $\sigma_{\mathcal{I}}$. For $\mathcal{I} = \emptyset$, define $\mathbf{N}_{\mathcal{I}}(\mathbf{x}) = [0, 0, \dots, 0]^T = \mathbf{0} \in \mathbb{R}^q$.

Remark 1: Any bijective map $\sigma_{\mathcal{I}}: \mathcal{I}_s \rightarrow \mathcal{I}$ suffices. For example, we can take the ascending order map defined recursively by $\sigma_{\mathcal{I}}(i) = \min(\mathcal{I} \setminus \cup_{j=1}^{i-1} \{\sigma_{\mathcal{I}}(j)\})$. A different bijective map only results in a rearrangement of the columns of the resulting matrix $\mathbf{N}_{\mathcal{I}}(\mathbf{x})$. Our final matrices of interest, $\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x})$ and $\mathbf{P}_{\mathcal{I}}(\mathbf{x})$ defined below, will be invariant with respect to such rearrangements.

For a full rank $\mathbf{N}_{\mathcal{I}}(\mathbf{x})$, define the two symmetric $q \times q$ projection matrices [5]

$$\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x}) = \begin{cases} \mathbf{0}, & \text{if } \mathcal{I} = \emptyset, \\ \mathbf{N}_{\mathcal{I}}(\mathbf{N}_{\mathcal{I}}^T \mathbf{N}_{\mathcal{I}})^{-1} \mathbf{N}_{\mathcal{I}}^T(\mathbf{x}), & \text{if } 1 \leq |\mathcal{I}| < q, \\ \mathbf{I}, & \text{otherwise.} \end{cases}$$

$$\mathbf{P}_{\mathcal{I}}(\mathbf{x}) = \mathbf{I} - \tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x}).$$

It was shown in [5, Lemma 1 and Theorem 1] that $\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x})$ takes any $\mathbf{z} \in \mathbb{R}^q$ to the subspace spanned by $\nabla h_i(\mathbf{x}), i \in \mathcal{I}$, and $\mathbf{P}_{\mathcal{I}}(\mathbf{x})$ takes \mathbf{z} into its orthogonal complement. In other words, for any $\mathbf{z} \in \mathbb{R}^q$, $\mathbf{P}_{\mathcal{I}}(\mathbf{x})\mathbf{z}$ will be parallel to the intersection of all supporting hyperplanes $H_i(\mathbf{x})$ whose indices are in \mathcal{I} .

Let the set of indices of all active constraints be

$$\mathcal{I}_{ac} = \{i \in \mathcal{I}_k \mid h_i(\mathbf{x}) \geq 0\},$$

where its dependence on \mathbf{x} has been dropped. Let \mathcal{J} be the set of all subsets of \mathcal{I}_{ac} with cardinality less than or equal to q . For a fixed $\mathbf{x} \in \mathbb{R}^q$, define the following *combinatorial optimization sub-problem*

$$\begin{aligned} & \max_{\mathcal{I} \in \mathcal{J}} \|\mathbf{P}_{\mathcal{I}}(\mathbf{x})\nabla J(\mathbf{x})\|, \\ \text{subject to} & \quad \text{rank}(\mathbf{N}_{\mathcal{I}}(\mathbf{x})) = |\mathcal{I}|, \\ & \mathbf{N}_{\mathcal{I}_{ac} \setminus \mathcal{I}}^T(\mathbf{x})\mathbf{P}_{\mathcal{I}}(\mathbf{x})\nabla J(\mathbf{x}) \geq \mathbf{0}, \end{aligned} \quad (5)$$

where the norm in the objective function is the usual Euclidean norm for \mathbb{R}^q . In words, sub-problem (5) is to find a subset of \mathcal{I}_{ac} such that the supporting hyperplanes whose indices are in this subset are linearly independent, the projection of $-\nabla J(\mathbf{x})$ onto the intersection of these hyperplanes is maximal in magnitude, and when \mathbf{x} is evolved in the resultant direction $-\mathbf{P}_{\mathcal{I}}(\mathbf{x})\nabla J(\mathbf{x})$, no constraints will be violated. With reference to Fig. 1, observe that the optimal solution at the point \mathbf{x}_2 is $\mathcal{I}^* = \{2\}$, as desired.

At each fixed time, let \mathcal{I}^* be the solution of sub-problem (5). The generalized continuous-time gradient projection method for problem (4) that can incorporate multiple

nonlinear constraints is then given by the update

$$\dot{\mathbf{x}} = -\mathbf{P}_{\mathcal{I}^*}(\mathbf{x})\nabla J(\mathbf{x}).$$

A. Generalized Scaled Continuous-time Gradient Projection Method

Similar to [7, Appendix B], the generalized *scaled* continuous-time gradient projection method can be obtained from the above by setting $\mathbf{x} = \mathbf{\Gamma}_1 \bar{\mathbf{x}}$, where $\mathbf{\Gamma}_1 \in \mathbb{R}^{q \times q}$ is a nonsingular constant scaling matrix. In this new coordinate system, the corresponding constrained nonlinear programming problem (4) becomes

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^q} J(\mathbf{\Gamma}_1 \bar{\mathbf{x}}), \quad \text{subject to } \mathbf{h}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) \leq \mathbf{0}, \quad (6)$$

with the associated gradient vectors

$$\begin{aligned} \nabla_{\bar{\mathbf{x}}} J(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) &= \mathbf{\Gamma}_1^T \nabla J(\mathbf{\Gamma}_1 \bar{\mathbf{x}}), \\ \nabla_{\bar{\mathbf{x}}} h_i(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) &= \mathbf{\Gamma}_1^T \nabla h_i(\mathbf{\Gamma}_1 \bar{\mathbf{x}}), \quad i \in \mathcal{I}_k. \end{aligned}$$

Then the transformed matrix $\tilde{\mathbf{N}}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}})$ is

$$\tilde{\mathbf{N}}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) = \mathbf{\Gamma}_1^T \mathbf{N}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}).$$

For a full rank $\mathbf{N}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}})$, the corresponding projection matrix $\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}})$ is

$$\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) = \begin{cases} \mathbf{I}, & \text{if } \mathcal{I} = \emptyset, \\ \mathbf{I} - \mathbf{\Gamma}_1^T \mathbf{Q}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) \mathbf{\Gamma}_1, & \text{if } 1 \leq |\mathcal{I}| < q, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

where for $\mathbf{\Gamma} = \mathbf{\Gamma}_1 \mathbf{\Gamma}_1^T \in \mathbb{R}^{q \times q}$,

$$\mathbf{Q}_{\mathcal{I}}(\mathbf{\Gamma}_1 \bar{\mathbf{x}}) = \mathbf{N}_{\mathcal{I}}(\mathbf{N}_{\mathcal{I}}^T \mathbf{\Gamma} \mathbf{N}_{\mathcal{I}})^{-1} \mathbf{N}_{\mathcal{I}}^T(\mathbf{\Gamma}_1 \bar{\mathbf{x}}).$$

Written in terms of \mathbf{x} , sub-problem (5) then becomes

$$\begin{aligned} & \max_{\mathcal{I} \in \mathcal{J}} \|\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x})\nabla_{\bar{\mathbf{x}}} J(\mathbf{x})\|, \\ \text{subject to} & \quad \text{rank}(\mathbf{N}_{\mathcal{I}}(\mathbf{x})) = |\mathcal{I}|, \\ & \tilde{\mathbf{N}}_{\mathcal{I}_{ac} \setminus \mathcal{I}}^T(\mathbf{x})\tilde{\mathbf{P}}_{\mathcal{I}}(\mathbf{x})\nabla_{\bar{\mathbf{x}}} J(\mathbf{x}) \geq \mathbf{0}. \end{aligned} \quad (7)$$

At each fixed time, let \mathcal{I}^* be the solution of sub-problem (7). The generalized scaled continuous-time gradient projection method for problem (6) is then given by the update

$$\dot{\bar{\mathbf{x}}} = -\tilde{\mathbf{P}}_{\mathcal{I}^*}(\mathbf{\Gamma}_1 \bar{\mathbf{x}})\nabla_{\bar{\mathbf{x}}} J(\mathbf{\Gamma}_1 \bar{\mathbf{x}}).$$

In the original coordinates, this becomes

$$\dot{\mathbf{x}} = -\mathbf{\Gamma}_1 \tilde{\mathbf{P}}_{\mathcal{I}^*}(\mathbf{x})\nabla_{\bar{\mathbf{x}}} J(\mathbf{x}). \quad (8)$$

B. Solving the Combinatorial Optimization Sub-problem

It is clear that sub-problem (5) is a specialization of sub-problem (7) for the case $\mathbf{\Gamma}_1 = \mathbf{I}$. For a finite number of active constraints $|\mathcal{I}_{ac}| = t$, there is only a finite number of candidate solutions given by $\sum_{i=0}^{\min(q,t)} \binom{t}{i}$. Hence there always exist an exhaustive search algorithm that solves sub-problem (5) and (7) that terminates in a finite number of steps. The solvability of sub-problem (5) and (7) is thus not an issue. In fact, for controllers with a small number of outputs, and hence a small number of saturation constraints, it is possible to enumerate all possibilities of sub-problem (7) and do away with an algorithm altogether. This is the

approach taken in Section VII. We leave the development of an efficient algorithm to solve sub-problem (7) as future work to focus on the main theme of this paper, which is the GPAW scheme.

VI. GRADIENT PROJECTION ANTI-WINDUP FOR NONLINEAR MIMO CONTROLLERS

In accordance with the classical interpretation that windup arises from an inconsistency between the controller state and output, the GPAW scheme aims to modify the controller state to keep $\|\text{sat}(\mathbf{u}_{\text{aw}}) - \mathbf{u}_{\text{aw}}\|$ as small as possible. For simplicity, we retain the entire controller structure of (2) and only design \mathbf{f}_{aw} in (3) to modify the way the controller state evolves. The GPAW compensated controller then has the form

$$\begin{aligned} \dot{\mathbf{x}}_{\text{aw}} &= \mathbf{f}_{\text{aw}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), & \mathbf{x}_{\text{aw}}(0) &= \mathbf{x}_{c0}, \\ \mathbf{u}_{\text{aw}} &= \mathbf{g}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), \end{aligned} \quad (9)$$

where \mathbf{y} , \mathbf{r} , \mathbf{x}_{c0} , and \mathbf{g}_c are identical to that in (2), $\mathbf{x}_{\text{aw}} \in \mathbb{R}^q$ is the state of the controller, and $\mathbf{u}_{\text{aw}} \in \mathbb{R}^m$ is the controller output. Here, we describe the use of the generalized scaled continuous-time gradient projection method of Section V-A to derive \mathbf{f}_{aw} from \mathbf{f}_c in (2), which is the main theme of this paper.

To preserve the nominal system performance when no control saturation occur, we must enforce $\mathbf{f}_{\text{aw}} = \mathbf{f}_c$. When some controls saturate, we want \mathbf{f}_{aw} to be such that \mathbf{x}_{aw} evolves in a way to keep $\|\text{sat}(\mathbf{u}_{\text{aw}}) - \mathbf{u}_{\text{aw}}\|$ small. Since \mathbf{f}_c is designed to achieve some nominal system performance, we can regard it as a “good” candidate direction to evolve \mathbf{x}_{aw} . Therefore, under control saturation, we want \mathbf{f}_{aw} to be as close to \mathbf{f}_c as possible, while maintaining the feasibility of the $2m$ saturation constraints

$$\mathbf{h}_{\text{sat}}(\mathbf{x}_{\text{aw}}) = \begin{bmatrix} \mathbf{g}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) - \mathbf{u}_{\text{max}} \\ -\mathbf{g}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) + \mathbf{u}_{\text{min}} \end{bmatrix} \leq \mathbf{0} \in \mathbb{R}^{2m}. \quad (10)$$

In Section V-A, the generalized scaled continuous-time gradient projection method was developed to solve the constrained nonlinear program (6). Even so, we can use just the underlying idea of gradient projection for anti-windup compensation, without actually solving an optimization problem. From (8), when no constraints are active, i.e. $\mathcal{I}_{\text{ac}} = \emptyset \Rightarrow \mathcal{I}^* = \emptyset$, we have the update

$$\dot{\mathbf{x}} = -\Gamma \nabla J(\mathbf{x}).$$

We can regard $-\Gamma \nabla J(\mathbf{x})$ as some nominal direction to drive $\dot{\mathbf{x}}$. When $\mathcal{I}_{\text{ac}} \neq \emptyset$, (8) gives the continuous-time update for $\dot{\mathbf{x}}$ that maintains the feasibility of the constraints in (6) while keeping the update direction “close” to $-\Gamma \nabla J(\mathbf{x})$. It can be seen that this is exactly what is needed to design \mathbf{f}_{aw} from the criteria listed above.

We can relate the quantities in Section V-A with those in (2), (9), and (10) by $\mathbf{x} \sim \mathbf{x}_{\text{aw}}$,

$$\mathbf{h}(\mathbf{x}) \sim \mathbf{h}_{\text{sat}}(\mathbf{x}_{\text{aw}}), \quad -\Gamma \nabla J(\mathbf{x}) \sim \mathbf{f}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}).$$

Observing that Γ is a chosen (invertible) symmetric positive definite matrix, the last relation gives

$\nabla J(\mathbf{x}) \sim -\Gamma^{-1} \mathbf{f}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r})$. The relations in Section V-A can now be rewritten using the above identifications. Let $\mathbf{h}_{\text{sat}}(\mathbf{x}_{\text{aw}})$ be decomposed as $\mathbf{h}_{\text{sat}}(\mathbf{x}_{\text{aw}}) = [h_{\text{sat}1}(\mathbf{x}_{\text{aw}}), h_{\text{sat}2}(\mathbf{x}_{\text{aw}}), \dots, h_{\text{sat}2m}(\mathbf{x}_{\text{aw}})]^T$. For any non-empty set of indices $\mathcal{I} \subset \mathcal{I}_{2m}$, $|\mathcal{I}| = s > 0$, and for some fixed $(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r})$, the corresponding $q \times s$ matrix $\mathbf{N}_{\mathcal{I}}(\mathbf{x}_{\text{aw}})$ is

$$\mathbf{N}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}) = [\nabla_{\mathbf{x}_{\text{aw}}} h_{\text{sat}\sigma_{\mathcal{I}(1)}}(\mathbf{x}_{\text{aw}}), \dots, \nabla_{\mathbf{x}_{\text{aw}}} h_{\text{sat}\sigma_{\mathcal{I}(s)}}(\mathbf{x}_{\text{aw}})]$$

where $\sigma_{\mathcal{I}}: \mathcal{I}_s \rightarrow \mathcal{I}$ is any chosen bijection as described in Section V. For $\mathcal{I} = \emptyset$, define $\mathbf{N}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}) = \mathbf{0} \in \mathbb{R}^q$. For a full rank $\mathbf{N}_{\mathcal{I}}(\mathbf{x}_{\text{aw}})$, define

$$\mathbf{f}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) = \begin{cases} \mathbf{f}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), & \text{if } \mathcal{I} = \emptyset, \\ \mathbf{R}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}) \mathbf{f}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), & \text{if } 1 \leq |\mathcal{I}| < q, \\ \mathbf{0}, & \text{otherwise,} \end{cases}$$

where

$$\mathbf{R}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}) = \mathbf{I} - \Gamma \mathbf{N}_{\mathcal{I}} (\mathbf{N}_{\mathcal{I}}^T \Gamma \mathbf{N}_{\mathcal{I}})^{-1} \mathbf{N}_{\mathcal{I}}^T(\mathbf{x}_{\text{aw}}).$$

Define the set of indices corresponding to active saturation constraints as

$$\mathcal{I}_{\text{sat}} = \{i \in \mathcal{I}_{2m} \mid h_{\text{sat}i}(\mathbf{x}_{\text{aw}}) \geq 0\}.$$

Let \mathcal{J} be the set of all subsets of \mathcal{I}_{sat} with cardinality less than or equal to q . Using $\max_{\mathcal{I} \in \mathcal{J}} \|\bar{\mathbf{P}}_{\mathcal{I}}(\mathbf{x}) \nabla_{\bar{\mathbf{x}}} J(\mathbf{x})\|^2$ as the equivalent objective for sub-problem (7), we get after simplification, the following combinatorial optimization sub-problem for some fixed $(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r})$

$$\begin{aligned} & \max_{\mathcal{I} \in \mathcal{J}} \mathbf{f}_c^T(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) \Gamma^{-1} \mathbf{f}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), \\ & \text{subject to} \quad \text{rank}(\mathbf{N}_{\mathcal{I}}(\mathbf{x}_{\text{aw}})) = |\mathcal{I}|, \\ & \quad \mathbf{N}_{\mathcal{I}_{\text{sat}} \setminus \mathcal{I}}^T(\mathbf{x}_{\text{aw}}) \mathbf{f}_{\mathcal{I}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) \leq \mathbf{0}. \end{aligned} \quad (11)$$

At each fixed time, with $(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r})$ fixed, let \mathcal{I}^* be the solution of sub-problem (11). Then the GPAW compensated controller derived from (2) is given by (9) with

$$\mathbf{f}_{\text{aw}}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) = \mathbf{f}_{\mathcal{I}^*}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}).$$

Note that the only tuning parameter for the GPAW scheme is the symmetric positive definite matrix Γ .

VII. APPLICATION TO ADAPTIVE CONTROL

The model of a nonlinear two-link robot is given by [12, pp. 396]

$$\begin{aligned} H_{11}(q_2) \ddot{q}_1 + H_{12}(q_2) \ddot{q}_2 - h(q_2) (2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) &= u_1, \\ H_{12}(q_2) \ddot{q}_1 + a_2 \ddot{q}_2 + h(q_2) \dot{q}_1^2 &= u_2, \end{aligned}$$

where

$$\begin{aligned} H_{11}(q_2) &= a_1 + 2a_3 \cos q_2 + 2a_4 \sin q_2, \\ H_{12}(q_2) &= a_2 + a_3 \cos q_2 + a_4 \sin q_2, \\ h(q_2) &= a_3 \sin q_2 - a_4 \cos q_2, \end{aligned}$$

$(a_1, a_2, a_3, a_4) = (3.34, 0.97, 3\sqrt{3}/5, 0.6)$ are constant system parameters, $\mathbf{u} = [u_1, u_2]^T$ is the input torque, and $\mathbf{q} = [q_1, q_2]^T$, $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2]^T$ are the joint angles and angular velocities respectively. It is assumed that (a_1, a_2, a_3, a_4) are *unknown* parameters, and $(\mathbf{q}, \dot{\mathbf{q}})$ are available measurements.

In the absence of saturation constraints, an adaptive sliding mode controller [12, Section 9.2], [9] that achieves global asymptotic tracking of some reference signal \mathbf{q}_d is given by

$$\mathbf{u} = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_D\mathbf{s}, \quad \dot{\hat{\mathbf{a}}} = -\Phi\mathbf{Y}^T\mathbf{s}, \quad (12)$$

where

$$\mathbf{s} = \dot{\hat{\mathbf{q}}} + \Lambda\tilde{\mathbf{q}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r, \quad \tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d,$$

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda\tilde{\mathbf{q}} = [\dot{q}_{r1}, \dot{q}_{r2}]^T,$$

$\mathbf{Y} \in \mathbb{R}^{2 \times 4}$, with elements $(\mathbf{Y})_{i,j} = y_{ij}$ given by

$$y_{11} = \ddot{q}_{r1}, \quad y_{12} = \ddot{q}_{r2}, \quad y_{21} = 0, \quad y_{22} = \ddot{q}_{r1} + \ddot{q}_{r2},$$

$$y_{13} = (2\ddot{q}_{r1} + \ddot{q}_{r2}) \cos q_2 - (\dot{q}_2\dot{q}_{r1} + \dot{q}_1\dot{q}_{r2} + \dot{q}_2\dot{q}_{r2}) \sin q_2,$$

$$y_{14} = (2\ddot{q}_{r1} + \ddot{q}_{r2}) \sin q_2 + (\dot{q}_2\dot{q}_{r1} + \dot{q}_1\dot{q}_{r2} + \dot{q}_2\dot{q}_{r2}) \cos q_2,$$

$$y_{23} = \ddot{q}_{r1} \cos q_2 + \dot{q}_1\dot{q}_{r1} \sin q_2,$$

$$y_{24} = \ddot{q}_{r1} \sin q_2 - \dot{q}_1\dot{q}_{r1} \cos q_2,$$

$\hat{\mathbf{a}} \in \mathbb{R}^4$ is the vector of estimated parameters (or controller state), $\mathbf{K}_D \in \mathbb{R}^{2 \times 2}$, $\Phi \in \mathbb{R}^{4 \times 4}$, $\Lambda \in \mathbb{R}^{2 \times 2}$ are chosen constant symmetric positive definite matrices, and $(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$ is assumed to be available for feedback.

We can identify (12) with (2) by

$$\mathbf{x}_c = \hat{\mathbf{a}}, \quad \mathbf{u}_c = \mathbf{u}, \quad \mathbf{y} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T, \quad \mathbf{r} = [\mathbf{q}_d^T, \dot{\mathbf{q}}_d^T, \ddot{\mathbf{q}}_d^T]^T,$$

$$\mathbf{f}_c(\mathbf{x}_c, \mathbf{y}, \mathbf{r}) = -\Phi\mathbf{Y}^T\mathbf{s}, \quad \mathbf{g}_c(\mathbf{x}_c, \mathbf{y}, \mathbf{r}) = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_D\mathbf{s}.$$

Let \mathbf{Y} be decomposed into its rows as $\mathbf{Y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$. The input-constrained system is characterized by the four saturation constraints, $\mathbf{h}_{\text{sat}}(\hat{\mathbf{a}}) = [h_{\text{sat}1}(\hat{\mathbf{a}}), h_{\text{sat}2}(\hat{\mathbf{a}}), h_{\text{sat}3}(\hat{\mathbf{a}}), h_{\text{sat}4}(\hat{\mathbf{a}})]^T \leq \mathbf{0}$,

$$h_{\text{sat}1}(\hat{\mathbf{a}}) = \mathbf{y}_1\hat{\mathbf{a}} - [1 \ 0] \mathbf{K}_D\mathbf{s} - u_{1\text{max}} \leq 0,$$

$$h_{\text{sat}2}(\hat{\mathbf{a}}) = \mathbf{y}_2\hat{\mathbf{a}} - [0 \ 1] \mathbf{K}_D\mathbf{s} - u_{2\text{max}} \leq 0,$$

$$h_{\text{sat}3}(\hat{\mathbf{a}}) = -\mathbf{y}_1\hat{\mathbf{a}} + [1 \ 0] \mathbf{K}_D\mathbf{s} + u_{1\text{min}} \leq 0,$$

$$h_{\text{sat}4}(\hat{\mathbf{a}}) = -\mathbf{y}_2\hat{\mathbf{a}} + [0 \ 1] \mathbf{K}_D\mathbf{s} + u_{2\text{min}} \leq 0,$$

which gives the gradient vectors

$$\nabla_{\hat{\mathbf{a}}} h_{\text{sat}1}(\hat{\mathbf{a}}) = \mathbf{y}_1^T, \quad \nabla_{\hat{\mathbf{a}}} h_{\text{sat}2}(\hat{\mathbf{a}}) = \mathbf{y}_2^T,$$

$$\nabla_{\hat{\mathbf{a}}} h_{\text{sat}3}(\hat{\mathbf{a}}) = -\mathbf{y}_1^T, \quad \nabla_{\hat{\mathbf{a}}} h_{\text{sat}4}(\hat{\mathbf{a}}) = -\mathbf{y}_2^T.$$

We apply the GPAW scheme to the above controller by enumerating all possibilities of sub-problem (11). After some tedious manipulations involving Boolean algebra, we arrive at the following anti-windup compensated controller

$$\dot{\mathbf{a}}_{\text{aw}} = \begin{cases} -\Phi\mathbf{Y}^T\mathbf{s}, & \text{if } A, \\ -\left(\mathbf{I} - \frac{1}{\mathbf{y}_1\Gamma\mathbf{y}_1^T}\Gamma\mathbf{y}_1^T\mathbf{y}_1\right)\Phi\mathbf{Y}^T\mathbf{s}, & \text{if } B, \\ -\left(\mathbf{I} - \frac{1}{\mathbf{y}_2\Gamma\mathbf{y}_2^T}\Gamma\mathbf{y}_2^T\mathbf{y}_2\right)\Phi\mathbf{Y}^T\mathbf{s}, & \text{if } C, \\ -\left(\mathbf{I} - \Gamma\mathbf{Y}^T(\mathbf{Y}\Gamma\mathbf{Y}^T)^{-1}\mathbf{Y}\right)\Phi\mathbf{Y}^T\mathbf{s}, & \text{otherwise,} \end{cases}$$

$$\mathbf{u}_{\text{aw}} = \mathbf{Y}\mathbf{a}_{\text{aw}} - \mathbf{K}_D\mathbf{s},$$

where

$$A_i = (h_{\text{sat}i}(\mathbf{a}_{\text{aw}}) \geq 0), \quad i \in \mathcal{I}_4,$$

$$f_1 = \mathbf{y}_1\Phi\mathbf{Y}^T\mathbf{s}, \quad B_1 = (f_1 \geq 0), \quad B_2 = (f_2 \geq 0),$$

$$f_2 = \mathbf{y}_2\Phi\mathbf{Y}^T\mathbf{s}, \quad B_3 = (f_1 \leq 0), \quad B_4 = (f_2 \leq 0),$$

$$g_1 = \mathbf{y}_2 \left(\mathbf{I} - \frac{1}{\mathbf{y}_1\Gamma\mathbf{y}_1^T}\Gamma\mathbf{y}_1^T\mathbf{y}_1 \right) \Phi\mathbf{Y}^T\mathbf{s},$$

$$g_2 = \mathbf{y}_1 \left(\mathbf{I} - \frac{1}{\mathbf{y}_2\Gamma\mathbf{y}_2^T}\Gamma\mathbf{y}_2^T\mathbf{y}_2 \right) \Phi\mathbf{Y}^T\mathbf{s},$$

$$g_3 = \mathbf{s}^T\mathbf{Y}\Phi \left(\frac{1}{\mathbf{y}_2\Gamma\mathbf{y}_2^T}\mathbf{y}_2^T\mathbf{y}_2 - \frac{1}{\mathbf{y}_1\Gamma\mathbf{y}_1^T}\mathbf{y}_1^T\mathbf{y}_1 \right) \Phi\mathbf{Y}^T\mathbf{s},$$

$$C_1 = (\text{rank}(\mathbf{y}_1) = 1), \quad C_2 = (\text{rank}(\mathbf{y}_2) = 1),$$

$$C_3 = (g_1 \geq 0), \quad C_4 = (g_1 \leq 0),$$

$$C_5 = (g_2 < 0), \quad C_6 = (g_2 > 0), \quad C_7 = (g_3 \geq 0),$$

$$A = ((\neg A_1 \wedge \neg A_3) \vee (A_1 \wedge B_1) \vee (A_3 \wedge B_3))$$

$$\wedge ((\neg A_2 \wedge \neg A_4) \vee (A_2 \wedge B_2) \vee (A_4 \wedge B_4)),$$

$$B = \neg A \wedge C_1 \wedge (A_1 \vee A_3) \wedge \left((\neg A_2 \wedge \neg A_4) \right.$$

$$\left. \vee \left(((A_2 \wedge C_3) \vee (A_4 \wedge C_4)) \wedge (\neg C_2 \vee C_7) \vee (A_1 \wedge C_5) \vee (A_3 \wedge C_6) \right) \right),$$

$$C = \neg A \wedge \neg B \wedge C_2 \wedge (A_2 \vee A_4) \wedge \left((\neg A_1 \wedge \neg A_3) \right.$$

$$\left. \vee \left(((A_1 \wedge \neg C_5) \vee (A_3 \wedge \neg C_6)) \wedge (\neg C_1 \vee (A_2 \wedge \neg C_3) \vee (A_4 \wedge \neg C_4)) \right) \right),$$

with the same definitions of the remaining variables in (12), and $\Gamma \in \mathbb{R}^{4 \times 4}$ is a chosen symmetric positive definite matrix. Note that in the above, A_i, B_i for $i \in \mathcal{I}_4$, C_i for $i \in \mathcal{I}_7$ and A, B, C are logical statements. It is clear that the GPAW compensated controller is a *hybrid* controller that switches its state update law over arbitrary combinations of saturating controls.

The nominal controller gains are fixed at $\Lambda = \mathbf{I}$, $\mathbf{K}_D = 10\mathbf{I}$, $\Phi = \text{diag}([30, 1, 10, 10])$. For a fixed reference trajectory $\mathbf{q}_d(t) = \pi(1 - \cos 2\pi t)[\frac{1}{6}, \frac{1}{4}]^T$, five cases are simulated for each scenario

- 1) unconstrained nominal system.
- 2) constrained but uncompensated system.
- 3) constrained with ‘‘stop integration’’ rule. The controller state is then updated by

$$\dot{\mathbf{x}}_{\text{aw}} = \begin{cases} -\Phi\mathbf{Y}^T\mathbf{s}, & \text{if } \text{sat}(\mathbf{u}_{\text{aw}}) = \mathbf{u}_{\text{aw}}, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

- 4) constrained with GPAW compensation, $\Gamma = \Phi$.
- 5) constrained with GPAW compensation, $\Gamma = \mathbf{I}$.

In all scenarios, the control limits are set at $u_{i\text{max}} = -u_{i\text{min}} = u_{\text{lim}}$ for $i \in \{1, 2\}$. For the unconstrained nominal system, it was found that setting $u_{\text{lim}} = 180$ Nm results in no control saturation *after transients*. Fig. 2 shows the tracking errors and control signals for the case where $u_{\text{lim}} = 150$ Nm. In the bottom two plots for control signals, note that cases 2 and 5 are not shown to avoid clutter. It can be seen that only case 4 (GPAW compensation with $\Gamma = \Phi$) gives satisfactory performance under control saturation. To demonstrate the graceful performance degradation of case 4, six scenarios are simulated, corresponding to setting u_{lim} to

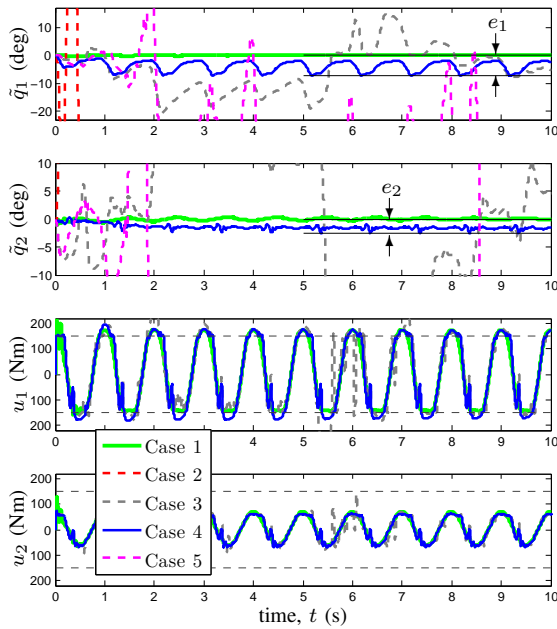


Fig. 2. Simulation results for $u_{\text{lim}} = 150$ Nm.

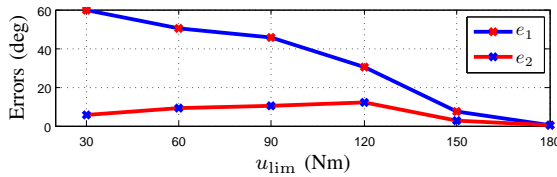


Fig. 3. Steady state tracking errors increase with saturation severity.

$\{180, 150, 120, 90, 60, 30\}$ Nm respectively. Observe that the last case represents approximately 17% control effectiveness, which will be considered severe by most standards. Fig. 3 shows the peak steady state tracking errors defined by $e_i = \max_{t \in [5, 10]} |\tilde{q}_i(t)|$ and illustrated in the top two plots of Fig. 2, as u_{lim} is varied. It can be seen that graceful performance degradation is exhibited for case 4 ($\Gamma = \Phi$).

VIII. LIMITATIONS OF GPAW

If the controller state and output are truly consistent, then we must achieve $\text{sat}(\mathbf{u}_{\text{aw}}) \equiv \mathbf{u}_{\text{aw}}$, i.e. \mathbf{u}_{aw} is maintained *exactly* at the saturation constraint boundaries when otherwise these same constraints would have been violated. In general, this will not be achieved. Consider now when $\tilde{\mathcal{I}}_{\text{sat}} \neq \emptyset$ is the set of indices of controller outputs that are saturated, $\tilde{\mathcal{I}}_{\text{sat}} \subset \mathcal{I}_m$, and (\mathbf{y}, \mathbf{r}) are such that they will worsen existing saturation conditions. Let \mathbf{g}_c of (9) be decomposed as $\mathbf{g}_c(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}) = [g_{c1}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), g_{c2}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r}), \dots, g_{cm}(\mathbf{x}_{\text{aw}}, \mathbf{y}, \mathbf{r})]^T$. Using the GPAW scheme, by [5, Theorem 3] we must have

$$\frac{\partial g_{ci}}{\partial \mathbf{x}_{\text{aw}}} \dot{\mathbf{x}}_{\text{aw}} = \nabla_{\mathbf{x}_{\text{aw}}} h_{\text{sat}i}^T(\mathbf{x}_{\text{aw}}) \dot{\mathbf{x}}_{\text{aw}} = 0, \quad i \in \tilde{\mathcal{I}}_{\text{sat}}.$$

To achieve $\text{sat}(\mathbf{u}_{\text{aw}}) \equiv \mathbf{u}_{\text{aw}}$, we must have $\dot{u}_{\text{awi}} = 0$ for $i \in \tilde{\mathcal{I}}_{\text{sat}}$. Expanding \dot{u}_{awi} , we get

$$\dot{u}_{\text{awi}} = \frac{\partial g_{ci}}{\partial \mathbf{x}_{\text{aw}}} \dot{\mathbf{x}}_{\text{aw}} + \frac{\partial g_{ci}}{\partial \mathbf{y}} \dot{\mathbf{y}} + \frac{\partial g_{ci}}{\partial \mathbf{r}} \dot{\mathbf{r}} = \frac{\partial g_{ci}}{\partial \mathbf{y}} \dot{\mathbf{y}} + \frac{\partial g_{ci}}{\partial \mathbf{r}} \dot{\mathbf{r}},$$

which will be non zero in general. Hence the GPAW scheme can only achieve consistency between the controller state and output at best in an approximate sense.

From the above, it can also be surmised that the GPAW scheme will be ineffective when

$$\left\| \frac{\partial \mathbf{g}_c}{\partial \mathbf{x}_c} \dot{\mathbf{x}}_c \right\| \ll \left\| \frac{\partial \mathbf{g}_c}{\partial \mathbf{y}} \dot{\mathbf{y}} + \frac{\partial \mathbf{g}_c}{\partial \mathbf{r}} \dot{\mathbf{r}} \right\|$$

holds for the constrained nominal system, since only the controller state will be modified.

IX. CONCLUSIONS

The generalized scaled continuous-time gradient projection method was developed and used for anti-windup compensation in the GPAW scheme, which can be viewed as a generalization of the “stop integration” heuristic. The key idea of the GPAW scheme is to project the controller state update law onto the tangent plane of all active saturation constraints in an attempt to maintain consistency between the controller state and output. Applying the GPAW scheme on some nominal controllers results in a derived *hybrid* controller that switches its state update law over arbitrary combinations of saturating controls. Simulations on a non-trivial example demonstrates its viability as a potential anti-windup scheme for nonlinear systems. The resultant hybrid controller was not obtained by deliberate design, and will provide the hybrid systems community with another meaningful real-world example for research purposes. Stability is currently being investigated for particular classes of nonlinear systems.

REFERENCES

- [1] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett, “A unified framework for the study of anti-windup designs,” *Automatica*, vol. 30, no. 12, pp. 1869 – 1883, Dec. 1994.
- [2] C. Edwards and I. Postlethwaite, “Anti-windup and bumpless-transfer schemes,” *Automatica*, vol. 34, no. 2, pp. 199 – 210, Feb. 1998.
- [3] S. Tarbouriech and M. Turner, “Anti-windup design: an overview of some recent advances and open problems,” *IET Control Theory Appl.*, vol. 3, no. 1, pp. 1 – 19, Jan. 2009.
- [4] F. Morabito, A. R. Teel, and L. Zaccarian, “Nonlinear antiwindup applied to Euler-Lagrange systems,” *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 526 – 537, Jun. 2004.
- [5] J. B. Rosen, “The gradient projection method for nonlinear programming. part I. linear constraints,” *J. Soc. Ind. Appl. Math.*, vol. 8, no. 1, pp. 181 – 217, Mar. 1960.
- [6] —, “The gradient projection method for nonlinear programming. part II. nonlinear constraints,” *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 514 – 532, Dec. 1961.
- [7] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice Hall, 1996. [Online]. Available: http://www-ref.usc.edu/~ioannou/Robust_Adaptive_Control.htm
- [8] K. J. Åström and L. Rundqwist, “Integrator windup and how to avoid it,” in *Proc. American Control Conf.*, Pittsburgh, PA, Jun. 1989, pp. 1693 – 1698.
- [9] J.-J. Slotine and J. A. Coetsee, “Adaptive sliding controller synthesis for non-linear systems,” *Int. J. Control*, vol. 43, no. 6, pp. 1631 – 1651, Jun. 1986.
- [10] A. S. Hodel and C. E. Hall, “Variable-structure PID control to prevent integrator windup,” *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 442 – 451, Apr. 2001.
- [11] J.-B. Pomet and L. Praly, “Adaptive nonlinear regulation: Estimation from the Lyapunov equation,” *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 729 – 740, Jun. 1992.
- [12] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Prentice Hall, 1991.