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

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>Version</td>
<td>Original manuscript</td>
</tr>
<tr>
<td>Accessed</td>
<td>Fri Dec 28 10:55:19 EST 2018</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/54789">http://hdl.handle.net/1721.1/54789</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher’s policy and may be subject to US copyright law. Please refer to the publisher’s site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>


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 \( f: (x, u) \rightarrow f(x, u) \), its partial derivative \( \frac{\partial f}{\partial x} \) is a row vector, and its gradient \( \nabla_x f = \left( \frac{\partial f}{\partial x} \right)^T \) is a column vector. For inequalities involving vectors, the inequality is to be interpreted element-wise. Let \( A \) and \( B \) be...
logical statements that evaluate to true or false. Then \(\neg A\), \(A \land B\) and \(A \lor B\) correspond to the statements “NOT A”, “A AND B”, and “A OR B” respectively. Let \(I\) and \(J\) be two sets. The cardinality of \(I\) will be denoted by \(|I|\), and \(I \setminus J\) is the relative complement of \(J\) in \(I\).

II. PROBLEM STATEMENT

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

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

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

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

where its dependence on \(u_{\text{min}}\) and \(u_{\text{max}}\) has been dropped. Consider the input constrained system described by

\[
\begin{aligned}
  \dot{x} &= f(x, \text{sat}(u)), \quad x(0) = x_0, \\
y &= g(x, \text{sat}(u)),
\end{aligned}
\]

and the nominal controller described by

\[
\begin{aligned}
  \dot{x}_c &= f_c(x_c, y, r(t)), \quad x_c(0) = x_{c0}, \\
y_c &= g_c(x_c, y, r(t)),
\end{aligned}
\]

where \(x, x_0 \in \mathbb{R}^n, u \in \mathbb{R}^m\), are the state, initial state and control input of system (1), \(y \in \mathbb{R}^p\) is the output of system (1) and measurement for the controller (2), \(x_c, x_{c0} \in \mathbb{R}^q, r(t) \in \mathbb{R}^r, y_c \in \mathbb{R}^m\), 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 \([0, \infty) \to \mathbb{R}^r\). Assume that the functions \(f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n, g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p, f_c : \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^q, \) and \(g_c : \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^m\), are such that the uncompensated closed loop system defined by (1), (2) and

\[
u = u_c,
\]

is well-posed, ie. existence and uniqueness of solutions are assured for all \(t \geq 0, (r, x_0, x_{c0}) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^q\), 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

\[
\begin{aligned}
  \dot{x}_\text{aw} &= f_\text{aw}(x_\text{aw}, y, r(t)), \quad x_\text{aw}(0) = x_\text{aw0}, \\
u_\text{aw} &= g_\text{aw}(x_\text{aw}, y, r(t)),
\end{aligned}
\]

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

\[
u = u_\text{aw}.
\]

The problem is to design \(f_\text{aw} : \mathbb{R}^\hat{q} \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^q\) and \(g_\text{aw} : \mathbb{R}^\hat{q} \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^m\), and determine an initialization \(x_\text{aw0} \in \mathbb{R}^\hat{q}\) such that:

(i) for every \((r, x_0, x_{c0}) \in \mathcal{R} \times \mathbb{R}^n \times \mathbb{R}^q\) such that the controls never saturate for the uncompensated system, ie. \(\text{sat}(u_c) \equiv u_c\), the control signal of the anti-windup compensated system satisfy \(u_\text{aw} \equiv u_c\). In other words, nominal system performance is recovered whenever no controls saturate.

(ii) when \((r, x_0, x_{c0})\) are such that some controls saturate for the uncompensated nominal system, ie. \(\text{sat}(u_c(t)) \neq u_c(t)\), 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 GPW 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 \(\text{sat}(u_{ci}, u_{\text{imin}}, u_{\text{imax}}) \neq u_{ci}\).

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

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

where

\[
A_i = \left( (u_{ci} \geq u_{\text{imax}}) \land \left( \frac{\partial g_{ci}}{\partial x_{ci}} f_{ci}(x_{ci}, y_i, r_i) > 0 \right) \right) \lor \left( (u_{ci} \leq u_{\text{imin}}) \land \left( \frac{\partial g_{ci}}{\partial x_{ci}} f_{ci}(x_{ci}, y_i, r_i) < 0 \right) \right).
\]

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

\[
\min_{x \in \mathbb{R}^n} J(x), \quad \text{subject to } h(x) \leq 0, \tag{4}
\]

where \(x \in \mathbb{R}^n\), \(J(x)\) is a possibly nonlinear scalar function, and \(h(x) = [h_1(x), h_2(x), \ldots, h_k(x)]^T\) 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 \(J(x)\) and \(h_i(x), i \in \{1, 2, \ldots, k\}\). 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 \(h_i(x) \leq 0, i \in \{1, 2, \ldots, k\}\), defines a hypersurface \(G_i\) in \(\mathbb{R}^n\) that forms the boundary of the feasible region \(S = \{x \in \mathbb{R}^n | h(x) \leq 0\}\). On each point \(x\) on the boundary of \(S\), each hypersurface \(G_i\) that contains \(x\) has an associated supporting hyperplane \(H_i(x)\) that is tangent to \(G_i\) at \(x\). The normal of \(H_i\) at the point \(x\) is the gradient of \(h_i(x)\) at \(x\), which will point “away” from \(S\). These are illustrated in Fig. 1.

Similar to many optimization methods, the gradient projection method generates a sequence \(\{x_n\}\), the limiting point of which would be the solution to the nonlinear program (4). Consider now, the case where all \(h_i(x)\) are affine functions of \(x\) [5]. Then \(G_i\) coincides with \(H_i\), and the boundary of \(S\) are all hyperplanes. At a particular point \(x_n\) that lies in the interior of \(S\) (cf. \(x_0\) in Fig. 1), the basic step is taken in a direction to decrease \(J(x_n)\), i.e. in the negative gradient direction, \(-\nabla J(x_n)\), much like the steepest descent method. When \(x_n\) lies on the boundary of \(S\) (cf. \(x_1\) and \(x_2\) in Fig. 1), the step is taken in a direction “closest” to \(-\nabla J(x_n)\) while remaining within \(S\). In this case, if \(-\nabla J(x_n)\) points into the interior of \(S\), the nominal direction is taken. Otherwise, \(-\nabla J(x_n)\) is projected onto the intersection of the smallest set of linearly independent hyperplanes \(H_i\) that corresponds to active constraints \((h_i(x_n) \geq 0)\) that can keep \(x_{n+1}\) within \(S\) (cf. \(x_1, x_2, x_3, x_4\) 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 \(x_{n+1}\) back to \(S\).

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 \(x_2\). Notice here that taking a step in the direction of \(-\nabla J(x_2)\) 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 \(z_2\). Notice that projecting onto \(H_1\) to get \(z_2\) 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 \(I_n = \{1, 2, \ldots, n\}\) where \(n\) is some positive integer. Then \(I_k\) 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 $s > 0$, i.e. $\mathcal{I} \neq \emptyset$, let $\sigma_\mathcal{I}: \mathcal{I}_s \rightarrow \mathcal{I}$ be a (non-unique) bijection map that assigns an integer in $\mathcal{I}$ to each integer in $\mathcal{I}_s$. For $s > 0$, define the $q \times s$ matrix

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

It can be seen that $\mathbf{N}_\mathcal{I}(x)$ 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 $\mathbf{N}_\mathcal{I}(x) = [0, 0, \ldots, 0]^T = \mathbf{0} \in \mathbb{R}^q$.

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

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

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

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

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

Let the set of indices of all active constraints be

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

where its dependence on $x$ 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 $x \in \mathbb{R}^q$, define the following combinatorial optimization sub-problem

$$
\max_{x \in \mathcal{J}} \|\mathbf{P}_\mathcal{I}(x) \nabla J(x)\|,
$$

subject to

$$
\mathbf{N}^T_{\mathcal{I}_{ac}}(x) \mathbf{P}_\mathcal{I}(x) \nabla J(x) \geq 0,
$$

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 $-\nabla J(x)$ onto the intersection of these hyperplanes is maximal in magnitude, and when $x$ is evolved in the resultant direction $-\mathbf{P}_\mathcal{I}(x) \nabla J(x)$, no constraints will be violated. With reference to Fig. 1, observe that the optimal solution at the point $x_2$ 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

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

### 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 $x = \Gamma_1 \bar{x}$, where $\Gamma_1 \in \mathbb{R}^{q \times q}$ is a nonsingular constant scaling matrix. In this new coordinate system, the corresponding constrained nonlinear programming problem (4) becomes

$$
\min_{\bar{x} \in \mathbb{R}^q} \ J(\Gamma_1 \bar{x}), \quad \text{subject to } h(\Gamma_1 \bar{x}) \leq 0,
$$

with the associated gradient vectors

$$
\nabla \bar{x} J(\Gamma_1 \bar{x}) = \Gamma_1^T \nabla J(\Gamma_1 \bar{x}),
$$

$$
\nabla \bar{x} h_i(\Gamma_1 \bar{x}) = \Gamma_1^T \nabla h_i(\Gamma_1 \bar{x}), \quad i \in \mathcal{I}_k.
$$

Then the transformed matrix $\tilde{\mathbf{N}}_\mathcal{I}(\Gamma_1 \bar{x})$ is

$$
\tilde{\mathbf{N}}_\mathcal{I}(\Gamma_1 \bar{x}) = \Gamma_1^T \mathbf{N}_\mathcal{I}(\Gamma_1 \bar{x}).
$$

For a full rank $\mathbf{N}_\mathcal{I}(\Gamma_1 \bar{x})$, the corresponding projection matrix $\mathbf{\tilde{P}}_\mathcal{I}(\Gamma_1 \bar{x})$ is

$$
\mathbf{\tilde{P}}_\mathcal{I}(\Gamma_1 \bar{x}) = \begin{cases} 
\mathbf{I}, & \text{if } \mathcal{I} = \emptyset, \\
\mathbf{I} - \Gamma_1^T \mathbf{Q}_\mathcal{I}(\Gamma_1 \bar{x}) \Gamma_1, & \text{if } 1 \leq |\mathcal{I}| < q, \\
\mathbf{0}, & \text{otherwise}.
\end{cases}
$$

where for $\Gamma = \Gamma_1 \Gamma_1^T \in \mathbb{R}^{q \times q}$,

$$
\mathbf{Q}_\mathcal{I}(\Gamma_1 \bar{x}) = \mathbf{N}_\mathcal{I} \left( \Gamma_1^T \mathbf{N}_\mathcal{I} \right)^{-1} \Gamma_1^T \mathbf{N}_\mathcal{I} (\Gamma_1 \bar{x}).
$$

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

$$
\max_{x \in \mathcal{J}} \|\mathbf{P}_\mathcal{I}(\Gamma_1 \bar{x}) \nabla \bar{x} J(\bar{x})\|,
$$

subject to

$$
\mathbf{N}^T_{\mathcal{I}_{ac}}(\bar{x}) \mathbf{\tilde{P}}_\mathcal{I}(\Gamma_1 \bar{x}) \nabla \bar{x} J(\bar{x}) \geq 0.
$$

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

$$
\dot{\bar{x}} = -\mathbf{P}_{\mathcal{I}^*}(\Gamma_1 \bar{x}) \nabla \bar{x} J(\Gamma_1 \bar{x}).
$$

In the original coordinates, this becomes

$$
\dot{x} = -\Gamma_1 \mathbf{P}_{\mathcal{I}^*}(x) \nabla J(x).
$$

### B. Solving the Combinatorial Optimization Sub-problem

It is clear that sub-problem (5) is a specialization of sub-problem (7) for the case $\Gamma_1 = \mathbf{I}$. 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 \(|\text{sat}(u_{aw}) - u_{aw}|\) as small as possible. For simplicity, we retain the entire controller structure of (2) and only design \(f_{aw}\) in (3) to modify the way the controller state evolves. The GPAW compensated controller then has the form

\[
\begin{aligned}
\dot{x}_{aw} &= f_{aw}(x_{aw}, y, r), \quad x_{aw}(0) = x_{c0}, \\
\dot{u}_{aw} &= g_{c}(x_{aw}, y, r),
\end{aligned}
\]

(9)

where \(y, r, x_{c0}\) and \(g_{c}\) are identical to that in (2), \(x_{aw} \in \mathbb{R}^q\) is the state of the controller, and \(u_{aw} \in \mathbb{R}^m\) is the controller output. Here, we describe the use of the generalized scaled continuous-time gradient projection method of Section V-A to derive \(f_{aw}\) from \(f_c\) in (2), which is the main theme of this paper.

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

\[
h_{\text{sat}}(x_{aw}) = \left[ g_{c}(x_{aw}, y, r) - u_{\text{max}} \right. \\
\left. - g_{c}(x_{aw}, y, r) + u_{\text{min}} \right] \preceq 0 \in \mathbb{R}^{2m}.
\]

(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. \(I_{\text{ac}} = \emptyset \Rightarrow I^* = \emptyset\), we have the update

\[
\dot{x} = -\Gamma \nabla J(x).
\]

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

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

\[
h(x) \sim h_{\text{sat}}(x_{aw}), \quad -\Gamma \nabla J(x) \sim f_c(x_{aw}, y, r).
\]

Observing that \(\Gamma\) is a chosen (invertible) symmetric positive definite matrix, the last relation gives \(\nabla J(x) \sim -\Gamma^{-1}f_c(x_{aw}, y, r)\). The relations in Section V-A can now be rewritten using the above identifications. Let \(h_{\text{sat}}(x_{aw})\) be decomposed as \(h_{\text{sat}}(x_{aw}) = [h_{\text{sat}1}(x_{aw}), h_{\text{sat}2}(x_{aw}), \ldots, h_{\text{sat}2m}(x_{aw})]^T\). For any non-empty set of indices \(I \subseteq I_{2m}, |I| = s > 0\), and for some fixed \((x_{aw}, y, r)\), the corresponding \(q \times s\) matrix \(N_I(x_{aw})\) is

\[
N_I(x_{aw}) = [\nabla_{x_{aw}} h_{\text{sat}1}(x_{aw}), \ldots, \nabla_{x_{aw}} h_{\text{sat}s}(x_{aw})]
\]

where \(\sigma_I: I_s \rightarrow I\) is any chosen bijection as described in Section V. For \(I = \emptyset\), define \(N_{\emptyset}(x_{aw}) = 0 \in \mathbb{R}^q\). For a full rank \(N_I(x_{aw})\), define

\[
f_I(x_{aw}, y, r) = \begin{cases} f_c(x_{aw}, y, r), & \text{if } I = \emptyset, \\
R_I(x_{aw})^{-1}f_c(x_{aw}, y, r), & \text{if } 1 \leq |I| < q, \\
0, & \text{otherwise}, \end{cases}
\]

where

\[
R_I(x_{aw}) = I - \Gamma N_I^T (N_I^2 \Gamma N_I)^{-1} N_I^T \Gamma N_I(x_{aw}).
\]

Define the set of indices corresponding to active saturation constraints as

\[
I_{\text{sat}} = \{ i \in I_{2m} \mid h_{\text{sat}1}(x_{aw}) \geq 0\}.
\]

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

\[
\max_{I \subseteq J} f_I^T(x_{aw}, y, r) \Gamma^{-1} f_I(x_{aw}, y, r),
\]

subject to \(\text{rank}(N_I(x_{aw})) = |I|\),

\[
N_I^T N_{I_{\text{sat}} \backslash I}(x_{aw}) I_{\text{sat}}(x_{aw}, y, r) \preceq 0.
\]

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

\[
f_{aw}(x_{aw}, y, r) = f_{I^*}(x_{aw}, y, r).
\]

Note that the only tuning parameter for the GPAW scheme is the symmetric positive definite matrix \(\Gamma\).

VII. APPLICATION TO ADAPTIVE CONTROL

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

\[
\begin{align*}
H_{11}(q_2)\dot{q}_1 + H_{12}(q_2)\dot{q}_2 - h(q_2) (2q_1\dot{q}_2 + \dot{q}_2^2) &= u_1, \\
H_{12}(q_2)\dot{q}_1 + a_2\dot{q}_2 + h(q_2)\dot{q}_1^2 &= u_2,
\end{align*}
\]

where

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

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

\[
u = Y \dot{a} - K_D s, \quad \dot{a} = -\Phi Y^T s, \tag{12}\]

where

\[
s = \hat{q} + \Lambda \dot{q} = \hat{q} - \dot{q}_r, \quad \ddot{q} = q - q_d.
\]

\[
\dot{q}_r = q_d - \Lambda \ddot{q} = [\ddot{q}_1, \ddot{q}_2]^T, \quad Y \in \mathbb{R}^{2 \times 4}, \text{ with elements } (Y)_{i,j} = y_{ij} \text{ given by}
\]

\[
y_{11} = \dot{q}_1, \quad y_{12} = \dot{q}_2, \quad y_{21} = 0, \quad y_{22} = \ddot{q}_1 + \dot{q}_2, 
\]

\[
y_{13} = (\dot{q}_1 + \dot{q}_2) \cos q_2 - (\dot{q}_2 \dot{q}_1 + \dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_1) \sin q_2,
\]

\[
y_{14} = (\dot{q}_1 + \dot{q}_2) \sin q_2 + (\dot{q}_2 \dot{q}_1 + \dot{q}_1 \dot{q}_2 + \dot{q}_2 \dot{q}_1) \cos q_2, 
\]

\[
y_{23} = \dot{q}_1 \cos q_2 + \dot{q}_2 \sin q_2, 
\]

\[
y_{24} = \ddot{q}_1 \sin q_2 - \dot{q}_1 \dot{q}_2 \cos q_2,
\]

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

We can identify (12) with (2) by

\[
x_c = \hat{a}, \quad u_c = u, \quad y = [q^T, \dot{q}^T]^T, \quad r = [q_d^T, \dot{q}_d^T, \ddot{q}_d^T]^T, 
\]

\[
f_c(x_c, y, r) = -\Phi Y^T s, \quad g_c(x_c, y, r) = Y \hat{a} - K_D s.
\]

Let \(Y\) be decomposed into its rows as \(Y = [y_1^T, y_2^T]^T\). The input-constrained system is characterized by the four saturation constraints, \(h_{sat}(\hat{a}) = [h_{sat1}(\hat{a}), h_{sat2}(\hat{a}), h_{sat3}(\hat{a}), h_{sat4}(\hat{a})]^T \leq 0\),

\[
h_{sat1}(\hat{a}) = y_1 \hat{a} - [1 \ 0] K_D s - u_{1\text{max}} \leq 0, \quad h_{sat2}(\hat{a}) = y_2 \hat{a} - [0 \ 1] K_D s - u_{2\text{max}} \leq 0, 
\]

\[
h_{sat3}(\hat{a}) = -y_1 \hat{a} + [1 \ 0] K_D s + u_{1\text{min}} \leq 0, \quad h_{sat4}(\hat{a}) = -y_2 \hat{a} + [0 \ 1] K_D s + u_{2\text{min}} \leq 0, 
\]

which gives the gradient vectors

\[
\nabla \hat{a} h_{sat1}(\hat{a}) = y_1^T, \quad \nabla \hat{a} h_{sat2}(\hat{a}) = y_2^T, 
\]

\[
\nabla \hat{a} h_{sat3}(\hat{a}) = -y_1^T, \quad \nabla \hat{a} h_{sat4}(\hat{a}) = -y_2^T.
\]

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 antiwindup compensated controller

\[
\dot{a}_{aw} = \begin{cases} -\Phi Y^T s, & \text{if } A, \\ (-I - \frac{1}{y_1 \Gamma_1} \hat{Y}_1 Y_1) \Phi Y^T s, & \text{if } B, \\ (-I - \frac{1}{y_2 \Gamma_2} \hat{Y}_2 Y_2) \Phi Y^T s, & \text{if } C, \\ (-I - \Gamma Y Y^T)^{-1} Y \Phi Y^T s, & \text{otherwise}, \end{cases}
\]

\[
u_{aw} = Y a_{aw} - K_D s,
\]

where

\[
A_i = (h_{sati}(a_{aw}) \geq 0), \quad i \in \mathcal{I}_4, 
\]

\[
f_1 = y_1 \Phi Y^T s, \quad B_1 = (f_1 \geq 0), \quad B_2 = (f_2 \geq 0),
\]

\[
f_2 = y_2 \Phi Y^T s, \quad B_3 = (f_1 \leq 0), \quad B_4 = (f_2 \leq 0),
\]

\[
g_1 = y_2 \left( I - \frac{1}{y_1 \Gamma_1} \hat{Y} Y_1 \right) \Phi Y^T s, 
\]

\[
g_2 = y_1 \left( I - \frac{1}{y_2 \Gamma_2} \hat{Y} Y_2 \right) \Phi Y^T s, 
\]

\[
g_3 = s^T Y \Phi \left( I - \frac{1}{y_2 \Gamma_2} \hat{Y} Y_2 \right) \Phi Y^T s
\]

\[
C_1 = (\text{rank}(y_1) = 1), \quad C_2 = (\text{rank}(y_2) = 1),
\]

\[
C_3 = (g_1 \geq 0), \quad C_4 = (g_1 \leq 0), \quad C_5 = (g_2 < 0), \quad C_6 = (g_2 > 0), \quad C_7 = (g_3 \geq 0),
\]

\[
A = ((\lnot A_1 \land \lnot A_3) \lor (A_1 \land A_3)) \lor (\lnot A_1 \land A_3)
\]

\[
\land ((\lnot A_2 \land \lnot A_4) \lor (A_2 \land A_4)) \lor (A_2 \land B_2) \lor (A_4 \land B_4),
\]

\[
B = (\lnot A \land C_1 \land (A_1 \lor A_3)) \lor (\lnot A \land A_3)
\]

\[
\land ((A_2 \land C_3) \lor (A_4 \land C_4)) \lor (\lnot C_2 \lor C_7)
\]

\[
\lor (A_1 \land C_5) \lor (A_3 \land C_6))
\]

\[
C = (\lnot A \land \lnot B \land C_2 \land (A_2 \lor A_4) \lor (\lnot A_1 \land \lnot A_3)
\]

\[
\land ((A_1 \land \lnot C_3) \lor (A_3 \land \lnot C_6)) \lor (\lnot C_1)
\]

\[
\lor (A_2 \land \lnot C_3) \lor (A_4 \land \lnot C_4))
\]

\[
\]

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 \(A = I, K_D = 10I, \Phi = \text{diag}([30, 1, 10, 10])\). For a fixed reference trajectory \(q_0(t) = \pi(1 - \cos 2\pi t)/2, [1, 1]^T\), 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

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

4) constrained with GPAW compensation, \(\Gamma = \Phi\),
5) constrained with GPAW compensation, \(\Gamma = I\).

In all scenarios, the control limits are set at \(u_{\text{imax}} = -u_{\text{imin}} = u_{\text{lim}}\) 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 \(u_{\text{lim}}\) to
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

$$\left| \frac{\partial g_c}{\partial \hat{x}_c} \right| \ll \left| \frac{\partial g_c}{\partial \hat{y}} \right| + \left| \frac{\partial g_c}{\partial \hat{r}} \right|$$

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


