



**HAL**  
open science

# Un tour d'horizon sur les techniques anti-windup pour les systèmes saturés

Sophie Tarbouriech, Isabelle Queinnec

► **To cite this version:**

Sophie Tarbouriech, Isabelle Queinnec. Un tour d'horizon sur les techniques anti-windup pour les systèmes saturés. Techniques de l'Ingénieur, 2020. hal-03070178

**HAL Id: hal-03070178**

**<https://laas.hal.science/hal-03070178>**

Submitted on 15 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un tour d'horizon sur les techniques anti-windup pour les systèmes saturés

La commande anti-windup

*An overview of anti-windup techniques for saturated systems - Anti-windup control*

par **Sophie TARBOURIECH**

Directrice de Recherche

LAAS, CNRS, Toulouse, France

par **Isabelle QUEINNEC**

Directrice de Recherche

LAAS, CNRS, Toulouse, France

## Résumé

Cet article traite de la synthèse anti-windup, technique de retouche d'un contrôleur qui permet de prendre en compte de manière explicite les phénomènes de saturation des actionneurs et de réduire leur impact négatif sur la dynamique d'un système commandé. Le procédé est décrit par un modèle d'état dynamique linéaire, de même que son contrôleur nominal construit a priori sans tenir compte des saturations. Les méthodes proposées s'appuient sur l'utilisation de conditions de secteurs afin d'encapsuler la fonction saturation et exploitent la stabilité de Lyapunov au travers de fonctions quadratiques permettant d'exprimer les conditions de stabilité et de performances par des inégalités matricielles linéaires.

## Abstract

Anti-windup design addressed in the paper is a controller reshaping technique allowing to explicitly take into account actuator saturation constraints and mitigate their effect on the closed-loop system dynamics. Both the process and the controller are described by linear state-space systems. The controller may be a priori designed without considering saturation elements. The anti-windup design approaches take advantage of sector conditions used to encapsulate the saturation function. Quadratic Lyapunov functions are used to generate linear matrix inequalities based stability and performance conditions.

## Mots-clés

saturations, actionneurs, contrôle/commande, anti-windup

## Keywords

saturations, actuators, control, anti-windup

# Table des matières

<b>1</b>	<b>Bref historique des techniques anti-windup</b>	<b>3</b>
<b>2</b>	<b>Systèmes saturés</b>	<b>4</b>
2.1	Systèmes considérés . . . . .	4
2.2	Stabilité des systèmes non-linéaires . . . . .	7
2.3	Outils mathématiques . . . . .	8
2.3.1	Fonctions de Lyapunov . . . . .	8
2.3.2	Approximation de la saturation (condition de secteur généralisée) . . . . .	9
<b>3</b>	<b>Architecture générale de l'anti-windup</b>	<b>10</b>
3.1	Principe . . . . .	10
3.2	Architectures de compensateurs anti-windup . . . . .	11
3.3	Formulation des problèmes . . . . .	12
<b>4</b>	<b>Réglage de correcteurs anti-windup</b>	<b>13</b>
4.1	Réglage d'anti-windup statique . . . . .	13
4.2	Réglage heuristique dans le cas SISO . . . . .	14
<b>5</b>	<b>Illustrations</b>	<b>14</b>
5.1	Exemple de synthèse d'un correcteur anti-windup . . . . .	14
5.2	Evaluation expérimentale d'un anti-windup statique pour un système monovarié . . . . .	17
<b>6</b>	<b>Extensions</b>	<b>20</b>

## Introduction

Un procédé commandé implique la mise à jour d'une entrée de commande, calculée en utilisant l'erreur entre une mesure de la sortie et sa consigne, au travers d'un actionneur. Dans le cas idéal, l'actionneur peut être représenté par un « fil » reliant la sortie du contrôleur et l'entrée du système. Dans la pratique, l'actionneur peut présenter un comportement dynamique, mais surtout une plage de fonctionnement limitée entre deux bornes, inférieure et supérieure. Ces limites sur la zone de fonctionnement linéaire de l'actionneur se traduisent par la notion de **saturation de l'actionneur**. Il s'agit d'un phénomène non-linéaire qui a comme effets indésirables de dégrader le comportement nominal du système, dans le meilleur des cas, voire de destabiliser le système contrôlé dans les cas extrêmes.

Pour éviter les effets indésirables de la saturation de l'actionneur, une solution classiquement utilisée est de surdimensionner l'actionneur, ce qui induit un coût supplémentaire tant du point de vue de la charge transportée, de l'énergie dépensée que du prix de l'actionneur lui-même. Une deuxième solution classique est de limiter les performances (rapidité) du contrôleur pour éviter, autant que possible, de s'approcher des limites de fonctionnement linéaire de l'actionneur.

La **commande anti-windup** est une alternative permettant d'utiliser au mieux les capacités de l'actionneur en autorisant les saturations tout en préservant la stabilité (et les performances) du système commandé dans la plage de fonctionnement prévue pour le système. Son principe est de réinjecter l'erreur commise par l'actionneur en entrée du système de commande. Cette retouche du contrôleur n'est active que lorsque l'actionneur est saturé, ce qui permet de garantir le comportement nominal du système commandé tel qu'il a été conçu originalement sans se préoccuper des non-linéarités de l'actionneur. Dans sa version la plus simple, la synthèse d'une commande anti-windup peut s'apparenter à la synthèse d'un simple gain supplémentaire, et ne présente pas plus de difficulté que la

synthèse d'un correcteur PID.

Par extension, le commande anti-windup s'applique aussi pour contrer les saturations en vitesse ou de dynamiques supérieures de l'actionneur, voire d'autres types de non-linéarités d'actionneurs, bien que ceci ne soit pas traité dans cet article.

*Nota* : Le lecteur trouvera en fin d'article un tableau des sigles, notations et symboles utilisés tout au long de l'article.

## 1 Bref historique des techniques anti-windup

Le terme "windup" est relié historiquement à l'effet de la saturation de l'actionneur dans une boucle de contrôle incluant une action intégrale [1], [2]. Dans ce cas, lorsque l'entrée sature le terme intégral continue à se charger alors que cela n'est pas nécessaire. La sortie de l'intégrateur est alors dite être en "windup". Cette charge non-nécessaire (sous la limite de la saturation) produit des effets indésirables sur la réponse transitoire du système tels que des oscillations autour du point d'équilibre ou bien encore une convergence lente au point d'équilibre. Ce type d'effets peut en particulier être observé lorsque le contrôleur a des modes lents ou des modes critiquement stables (pôles sur l'axe imaginaire). Afin d'éviter la dégradation de performance, voire la perte de la stabilité, générée par l'occurrence de la saturation, une boucle additionnelle de commande, appelée compensation *anti-windup*, peut être ajoutée au système en boucle fermée.

Le principe de base d'une stratégie anti-windup consiste à utiliser la différence entre le signal en sortie du contrôleur ( $y_c(t)$ ) et le signal effectivement appliqué au système à contrôler ( $sat(y_c(t))$ ). Cette différence est alors réinjectée dans le contrôleur via une boucle statique (un simple gain) ou une boucle dynamique (un compensateur dynamique). Cette nouvelle boucle est appelée *compensateur anti-windup*. Ce principe général est décrit dans la figure 1.

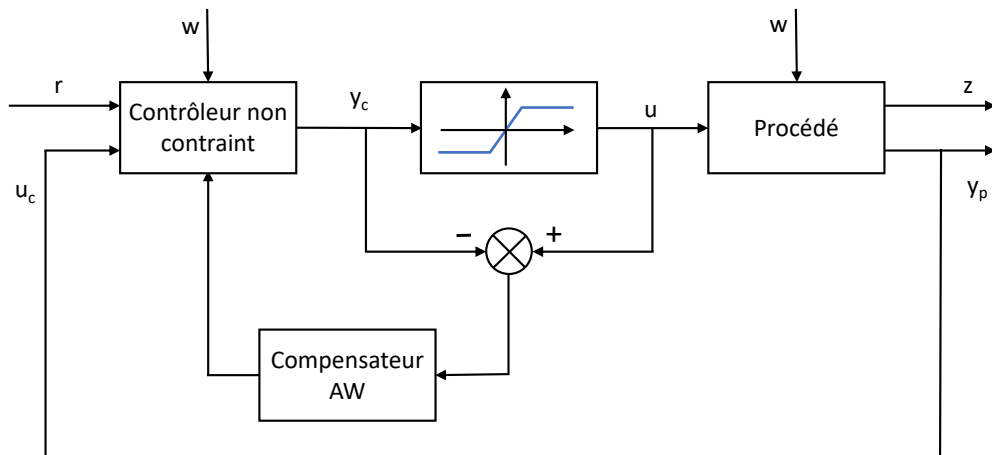


Figure 1 – Schéma de principe de l'architecture anti-windup

L'idée avec cette boucle additionnelle est de modifier (ou corriger) les états du contrôleur nominal (c'est-à-dire celui conçu sans tenir compte des saturations) lorsque la saturation est active. Il est important de remarquer qu'en l'absence de saturation on a  $sat(y_c(t)) - y_c(t) = 0$  et donc la boucle anti-windup n'est pas active, laissant ainsi le système bouclé se comporter avec les performances déterminées par le contrôleur nominal. C'est une des raisons qui fait que cette stratégie est très utilisée dans la pratique (voir par exemple [3], [4], [5], [6]).

L'objectif lorsque l'on synthétise une boucle anti-windup peut être bien sûr de garantir un certain niveau de performance pour le système en présence de saturation, et en particulier de diminuer la dégradation des performances linéaires (obtenues sans saturation) mais aussi de permettre d'agrandir la région d'attraction de l'origine (ou au moins son estimé) du système bouclé. Le lecteur peut consulter par exemple [7], [8], [9], [10], qui proposent des descriptions détaillées des techniques anti-windup ainsi qu'un tour d'horizon des problèmes associés.

## 2 Systèmes saturés

### 2.1 Systèmes considérés

**Nota :** Tout au long de l'article, la dépendance explicite au temps des variables est omise afin de simplifier les notations.

L'interconnexion d'un procédé avec un système de commande met en jeu des capteurs, pour transmettre les informations du procédé au contrôleur, et des actionneurs, pour transmettre la commande au procédé. Dans cet article, la problématique des capteurs n'est pas explicitement abordée, mais les résultats présentés peuvent généralement être étendus assez directement à la présence de non-linéarités au niveau de ceux-ci. De manière générale, les actionneurs sont des systèmes dynamiques comportant des contraintes, sous forme de saturations, tant en position que sur les dynamiques d'ordre supérieur. Ainsi, dans cet article, on considère que l'actionneur délivre un signal  $u$  prenant ses valeurs dans un ensemble compact, noté  $S(I_m, u_{min}, u_{max})$  et défini par :

$$S(I_m, u_{min}, u_{max}) = \{u \in \mathfrak{R}^m : -u_{min} \preceq u \preceq u_{max}\} \quad (1)$$

Le symbole  $\preceq$  signifie que l'inégalité est calculée élément par élément et  $u_{min}$  et  $u_{max}$  sont des vecteurs de  $\mathfrak{R}^m$  à éléments positifs, à savoir  $u_{min(i)} > 0$ ,  $u_{max(i)} > 0$ ,  $\forall i = 1, \dots, m$ . Dans la suite de cet article, seul est considéré le cas des saturations symétriques, en considérant  $u_{max} = u_{min} = u_0$ .

Dans la pratique, la présence de ces non-linéarités inhérentes à l'actionneur doit être prise en compte, car elles peuvent générer des dégradations du comportement attendu du système contrôlé, allant d'oscillations non désirées à la perte de stabilité selon les cas, ou imposent des restrictions sur la plage de fonctionnement du système [S7192]. Ceci est illustré sur l'exemple de motivation suivant.

#### Exemple 1 : Exemple de motivation

Soit le modèle instable d'un pointeur oscillant décrit dans l'ouvrage de Kailath [11]

$$\dot{x}_p = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x_p + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u = A_p x_p + B_{pu} u \quad (E1.1)$$

Le contrôleur  $K = \begin{bmatrix} 13 & 7 \end{bmatrix}$  stabilise globalement asymptotiquement l'origine du système en l'absence de bornes sur la commande (les valeurs propres de  $A_p + B_{pu}K$  sont -3 et -4).

Si l'on considère à présent des limitations de l'actionneur entre -5 et +5, le système en boucle fermée devient non-linéaire :

$$\dot{x} = A_p x + B_{pu} \text{sat}(Kx) \quad (E1.2)$$

$$\text{avec } \text{sat}(Kx_p) = \begin{cases} 5 & \text{si } Kx_p > 5 \\ Kx_p & \text{si } -5 \leq Kx_p \leq 5 \\ -5 & \text{si } Kx_p < -5 \end{cases}$$

Il existe dans ce cas 3 points d'équilibre, à savoir l'origine,  $\begin{bmatrix} -5 \\ 0 \end{bmatrix}$  et  $\begin{bmatrix} 5 \\ 0 \end{bmatrix}$ , qui sont obtenus en examinant les différentes valeurs possibles de l'entrée  $u$ . Le système n'est plus globalement asymptotiquement stable et, selon l'état initial, les trajectoires peuvent converger vers l'origine ou vers l'un des autres points d'équilibre, ou bien encore diverger vers l'infini.

De manière similaire, si une perturbation en amplitude  $w$  affecte le système pendant deux secondes en agissant sur l'entrée de commande ( $u$  est remplacé par  $u + w$ ), selon la valeur de  $w$ ,  $u$  peut saturer de manière transitoire, et la trajectoire du système bouclé peut diverger (par exemple pour  $w = 6$  dans le cas du système considéré dans cet exemple), bien que le système soit « BIBO-stable », c'est-à-dire satisfait la propriété « Entrée-bornée - Sortie-bornée ».

De manière générique, une représentation locale linéaire à temps invariant (LTI) du procédé à commander est considérée, sous la forme d'état standard :

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_{pu} u + B_{pw} w \\ y_p &= C_p x_p + D_{pu} u + D_{pw} w \\ z &= C_z x_p + D_{zu} u + D_{zw} w \end{aligned} \quad (2)$$

avec  $x_p \in \mathbb{R}^n$  l'état du procédé,  $u \in \mathbb{R}^m$  l'entrée de commande,  $w \in \mathbb{R}^q$  l'entrée de perturbation,  $y_p \in \mathbb{R}^p$  la sortie mesurée et  $z \in \mathbb{R}^l$  la sortie à contrôler. Les matrices  $A_p$ ,  $B_{pu}$ ,  $B_{pw}$ ,  $C_p$ ,  $C_z$ ,  $D_{pu}$ ,  $D_{pw}$ ,  $D_{zu}$  et  $D_{zw}$  sont des matrices constantes de dimensions appropriées et les paires  $(A_p, B_{pu})$  et  $(C_p, A_p)$  sont supposées contrôlables et observables, bien que ces hypothèses puissent être relâchées en stabilisabilité et détectabilité.

De manière similaire, les contrôleurs sont aussi décrits sous forme d'état :

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c u_c + B_{cw} w \\ y_c &= C_c x_c + D_c u_c + D_{cw} w \end{aligned} \quad (3)$$

avec  $x_c \in \mathbb{R}^{n_c}$  l'état du contrôleur,  $u_c \in \mathbb{R}^p$  l'entrée du contrôleur et  $y_c \in \mathbb{R}^m$  la sortie du contrôleur. Les dimensions des entrées et sorties du contrôleur sont considérées égales aux dimensions respectives des sorties et entrées du procédé. La présence d'allocateurs n'est pas abordée explicitement dans cet article [12], [13], mais ceux-ci pourraient être intégrés au contrôleur. Les matrices des contrôleurs  $(A_c, B_c, C_c, D_c)$ , de dimensions appropriées, sont supposées avoir été synthétisées pour garantir des propriétés pour le système en boucle fermée (stabilité, rapidité, rejet de perturbations...) dans le cas idéal sans saturation.  $B_{cw}$  et  $D_{cw}$  sont les matrices d'entrées de perturbations exogènes pouvant agir directement sur le contrôleur.

Différentes classes de perturbations exogènes peuvent être considérées, telles que les perturbations bornées en amplitude, qui permettent aussi de représenter des changements de référence, et des perturbations en énergie. Seules ces dernières sont considérées ici. Il s'agit de perturbations appartenant à un ensemble  $\mathcal{W}$  défini comme suit :

$$\mathcal{W} = \left\{ w : [0, \infty) \rightarrow \mathbb{R}^q ; \int_0^\infty w(\tau)^\top R w(\tau) d\tau \leq \delta^{-1} \right\} \quad (4)$$

et exprimant que l'énergie de la perturbation est bornée par  $\delta^{-1}$ ,  $\delta$  étant un scalaire positif. Plus précisément, cet article considère le cas où  $R$  (matrice symétrique définie positive) est en fait égale à la matrice identité (de dimension  $q$ ), ce qui fait que  $\mathcal{W}$  représente l'**ensemble des perturbations bornées- $\mathcal{L}_2$** .

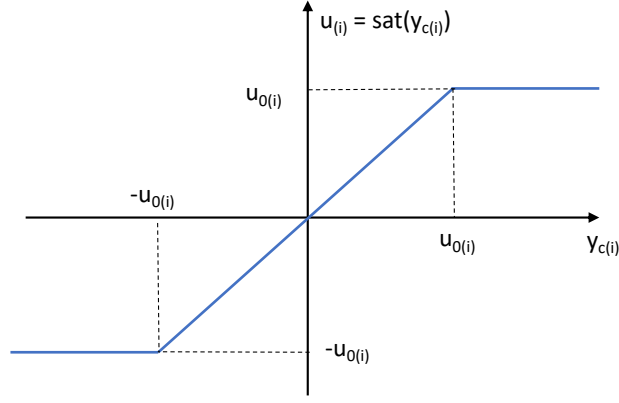


Figure 2 – Fonction saturation

L'interconnexion entre le procédé et son contrôleur est définie par les relations suivantes :

$$u_c = y_p \quad (5)$$

$$u_p = \text{sat}(y_c) \quad (6)$$

La fonction **saturation** reliant la sortie du contrôleur à l'entrée du procédé, illustrée sur la figure 2, est une fonction vectorielle définie, entre des bornes  $-u_0$  et  $u_0$  par

$$\text{sat}(y_{c(i)}) = \begin{cases} u_{0(i)} & \text{si } y_{c(i)} > u_{0(i)} \\ y_{c(i)} & \text{si } -u_{0(i)} \leq y_{c(i)} \leq u_{0(i)} \\ -u_{0(i)} & \text{si } y_{c(i)} < -u_{0(i)} \end{cases} \quad (7)$$

pour  $i = 1, \dots, m$ .

Une autre manière de prendre en compte la saturation de l'actionneur consiste à manipuler une zone morte, représentant l'écart entre la sortie et l'entrée de la saturation.

La fonction **zone morte**, illustrée sur la figure 3, est une fonction vectorielle définie, entre des bornes  $-u_0$  et  $u_0$  par

$$\phi(y_{c(i)}) = \begin{cases} u_{0(i)} - y_{c(i)} & \text{si } y_{c(i)} > u_{0(i)} \\ 0 & \text{si } -u_{0(i)} \leq y_{c(i)} \leq u_{0(i)} \\ -u_{0(i)} - y_{c(i)} & \text{si } y_{c(i)} < -u_{0(i)} \end{cases} \quad (8)$$

pour  $i = 1, \dots, m$ .

L'intérêt principal d'utiliser la fonction zone morte plutôt que la fonction saturation dans le système en boucle fermée est qu'elle permet d'exprimer celui-ci avec la dynamique linéaire de manière explicite, complétée de l'effet de la saturation qui intervient comme une entrée exogène. Il peut ainsi être décrit, en considérant l'état augmenté  $x_{bf} = \begin{bmatrix} x_p \\ x_c \end{bmatrix}$ , sous la forme générique suivante :

#### Forme générique du système en boucle fermée saturé

$$\begin{aligned} \dot{x}_{bf} &= \mathbb{A}x_{bf} + \mathbb{B}_\phi \phi(y_c) + \mathbb{B}_w w \\ y_c &= \mathbb{C}_y x_{bf} + \mathbb{D}_{y\phi} \phi(y_c) + \mathbb{D}_{yw} w \\ z &= \mathbb{C}_z x_{bf} + \mathbb{D}_{z\phi} \phi(y_c) + \mathbb{D}_{zw} w \end{aligned} \quad (9)$$

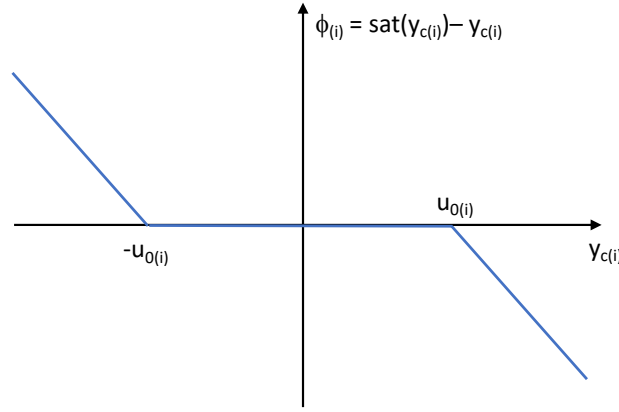


Figure 3 – Fonction zone morte

avec

$$\begin{aligned}
\mathbb{A} &= \begin{bmatrix} A_p + B_{pu}\Delta^{-1}D_cC_p & B_{pu}\Delta^{-1}C_c \\ B_c(I_p + D_{pu}\Delta^{-1}D_c)C_p & A_c + B_cD_{pu}\Delta^{-1}C_c \end{bmatrix} \\
\mathbb{B}_\phi &= \begin{bmatrix} B_{pu}(I_m + \Delta^{-1}D_cD_{pu}) \\ B_cD_{pu}(I_m + \Delta^{-1}D_cD_{pu}) \end{bmatrix} \\
\mathbb{B}_w &= \begin{bmatrix} B_{pu}\Delta^{-1}(D_{cw} + D_cD_{pw}) + B_{pw} \\ B_cD_{pu}\Delta^{-1}(D_{cw} + D_cD_{pw}) + B_{cw} + B_cD_{pw} \end{bmatrix} \\
\mathbb{C}_y &= \begin{bmatrix} \Delta^{-1}D_cC_p & \Delta^{-1}C_c \end{bmatrix}; \quad \mathbb{C}_z = \begin{bmatrix} C_z + D_{zu}\Delta^{-1}D_cC_p & D_{zu}\Delta^{-1}C_c \end{bmatrix} \\
\mathbb{D}_{y\phi} &= \Delta^{-1}D_cD_{pu}; \quad \mathbb{D}_{yw} = \Delta^{-1}(D_{cw} + D_cD_{pw}) \\
\mathbb{D}_{z\phi} &= D_{zu}(I_m + \Delta^{-1}D_cD_{pu}); \quad \mathbb{D}_{zw} = D_{zw} + D_{zu}\Delta^{-1}(D_{cw} + D_cD_{pw})
\end{aligned} \tag{10}$$

et  $\Delta = I_m - D_cD_{pu}$  avec l'hypothèse de bien-posé ( $\det(\Delta) \neq 0$ ).

**Nota :** Le cas particulier du retour d'état  $y_c = Kx_p$  s'écrit avec la même structure sous la forme  $\dot{x}_p = (A_p + B_{pu}K)x_p + B_{pu}\phi(Kx_p) + B_{pw}w$  avec  $\mathbb{A} = A_p + B_{pu}K$ ,  $\mathbb{B}_\phi = B_{pu}$ ,  $\mathbb{B}_w = B_{pw}$  et  $\mathbb{C}_y = K$ .

## 2.2 Stabilité des systèmes non-linéaires

Cet article n'a pas vocation à traiter de la théorie de la stabilité, mais il est important de rappeler quelques notions de base concernant l'analyse des systèmes dynamiques sur lesquelles s'appuie la suite de cet article. En particulier, toutes les méthodes qui suivent s'appuient sur la notion de stabilité au sens de Lyapunov, qui permet de conclure quant à la stabilité d'un système non-linéaire dans un voisinage autour de l'origine.

La présence de la saturation dans le système bouclé (9) implique de s'intéresser au comportement non-linéaire du système. Ainsi contrairement à un système linéaire, la convergence à l'origine (point d'équilibre d'intérêt lorsque  $w = 0$ ) dépend de la condition initiale. Même si la matrice  $\mathbb{A}$  est Hurwitz, en plus des conditions initiales  $x_{bf}(0)$  telles que les trajectoires convergent à l'origine, il peut exister d'autres conditions initiales pour lesquelles les trajectoires peuvent diverger ou converger vers d'autres points d'équilibre (voir Exemple 1). On appelle l'ensemble des conditions initiales pour lesquelles les trajectoires convergent vers l'origine, le bassin d'attraction de l'origine et on définit cette région de l'espace d'état comme :

$$R_A = \{x_{bf} \in \mathfrak{R}^{n_p+n_c}; x_{bf}(t, x_{bf}(0)) \rightarrow 0 \text{ quand } t \rightarrow \infty\} \tag{11}$$

Sous une certaine hypothèse de stabilité de la boucle ouverte (matrice  $A_p$ ) cette région peut être l'espace d'état :  $R_A = \mathfrak{R}^n$ , mais en général, la région  $R_A$  est un sous-ensemble de l'espace d'état :  $R_A \subset \mathfrak{R}^n$ . La région  $R_A$  peut être non-convexe, ouverte et non bornée, et sa caractérisation exacte est très difficile (voir par exemple [14], [15], [9], [10]). C'est pourquoi on s'intéresse à l'obtention



d'une estimation ou d'une approximation de cette région  $R_A$  via une région de stabilité, notée  $R_S$  et définie comme :

$$R_S \subseteq R_A \text{ et } 0 \in R_S \quad (12)$$

On pourra ainsi garantir que pour toute condition initiale appartenant à  $R_S$  les trajectoires du système bouclé convergeront (asymptotiquement) vers l'origine.

Considérer  $R_A$  et  $R_S$  implique de considérer principalement deux notions de stabilité propres aux systèmes non-linéaires : la stabilité (asymptotique) globale et la stabilité (asymptotique) locale. La stabilité globale est associée à la propriété que pour toute condition initiale de l'espace d'état, les trajectoires du système bouclé convergent à l'origine. Ceci correspond au cas  $R_A = \mathfrak{R}^n$ . Lorsqu'au contraire, on ne peut garantir la convergence des trajectoires à l'origine que pour un ensemble de conditions initiales on parle de stabilité locale. Ceci correspond au cas où la région  $R_A$  est plus petite que  $\mathfrak{R}^n$ .

De manière similaire, la présence de perturbation ( $w \neq 0$ ) dans le système (9) implique de s'intéresser au fait que les trajectoires restent bornées malgré l'effet de la perturbation, et convergeront vers l'origine si la perturbation est évanescence. De ces notions, on en déduit qu'étudier un système saturé implique de s'intéresser à son domaine (d'état) de fonctionnement, ses perturbations admissibles et ses performances (gain entre  $w$  et  $z$ ).

## 2.3 Outils mathématiques

A présent, il s'agit de caractériser une région  $R_S$  correspondant à une « bonne » estimation de la région  $R_A$ .

### 2.3.1 Fonctions de Lyapunov

Le principal outil utilisé est basé sur la théorie de Lyapunov et donc les fonctions de Lyapunov. L'intérêt de l'utilisation de la seconde méthode de Lyapunov est d'étudier la convergence des trajectoires à l'origine sans explicitement décrire ces trajectoires, mais en utilisant une fonction qui est signée (définie positive) dans une région de l'espace d'état incluant l'origine et nulle à l'origine. La dérivée de cette fonction doit être définie négative pour garantir la stabilité asymptotique. Ainsi s'il existe une fonction  $V : \mathcal{D} \rightarrow \mathcal{R}$ , continûment différentiable dans  $\mathcal{D}$  contenant l'origine, telle que

$$C1) \quad V(0) = 0 \text{ et } V(x) > 0, \forall x \in \mathcal{D}, x \neq 0;$$

$$C2) \quad \dot{V}(x) < 0, \forall x \in \mathcal{D}, x \neq 0,$$

alors, l'origine est asymptotiquement stable.

Ceci ne constitue qu'une condition suffisante pour garantir la stabilité de l'origine. La difficulté majeure réside dans la détermination d'une telle fonction  $V$ . On peut s'appuyer sur l'interprétation physique d'une fonction de Lyapunov comme par exemple dans les domaines de la mécanique ou des circuits électriques, avec la notion d'énergie.

Lorsque l'on a en main une fonction de Lyapunov pour le système bouclé (9) (c'est -à-dire satisfaisant les conditions C1) et C2)), une caractérisation naturelle d'une région de stabilité peut être obtenue à partir de :

$$S(V, c) = \{x_{bf} \in \mathfrak{R}^n; V(x_{bf}) \leq c^{-1}\}, c > 0 \quad (13)$$

La région  $S(V, c)$  est une région de stabilité asymptotique et est aussi un ensemble positivement invariant : c'est-à-dire pour toute condition initiale  $x_{bf}(0) \in S(V, c)$ , on a (dans le cas sans perturbation),  $x_{bf}(t, x_{bf}(0)) \in S(V, c)$  et  $x_{bf}(t, x_{bf}(0)) \rightarrow 0$  quand  $t \rightarrow \infty$ . Par ailleurs, ce type de

région permet aussi de caractériser la région dans laquelle les trajectoires du système bouclé restent bornées en présence de perturbation ( $w \neq 0$ ).

Comme précédemment mentionné, il peut être compliqué de trouver une fonction de Lyapunov adéquate. Dans ce qui suit, des fonctions quadratiques sont considérées, correspondant au cas où  $V(x)$  est définie comme :

$$V(x_{bf}) = x_{bf}^T P x_{bf}, \text{ avec } P = P^T > 0 \quad (14)$$

La région  $S(V, c)$  est alors un ellipsoïde que l'on redéfinit comme :

$$\mathcal{E}(P, c) = \{x_{bf} \in \mathbb{R}^n; x_{bf}^T P x_{bf} \leq c^{-1}\} \quad (15)$$

avec  $c > 0$ .

### 2.3.2 Approximation de la saturation (condition de secteur généralisée)

Même si la fonction saturation peut apparaître comme une simple non-linéarité, son utilisation pour construire des conditions constructives de stabilité et de performance nécessite d'utiliser des représentations plus appropriées à un traitement mathématique. Ces représentations sont généralement des sur-approximations de la saturation et ne permettent donc que de construire des conditions suffisantes pour les problèmes à résoudre. Cependant, ces conditions suffisantes peuvent être résolues numériquement et permettent l'analyse effective du système bouclé, ou la synthèse des contrôleurs. On peut aussi montrer que ces conditions sont moins complexes numériquement que d'autres approches de la littérature, et peuvent être moins conservatives. Il existe plusieurs représentations de la fonction saturation. Dans cet article, la saturation est encapsulée dans une condition de secteur, qui permet ensuite de traiter le problème d'analyse de stabilité comme un problème Lur'e [15]. Dans sa version généralisée, elle est définie comme suit, avec  $\phi(\cdot)$  la fonction zone morte introduite dans l'équation (8).

**Lemme :** Si les vecteurs  $\lambda_1$  et  $\lambda_2$  sont des éléments de l'ensemble  $S(\lambda_1 - \lambda_2, u_0) = \{\lambda_1 \in \mathbb{R}^m; \lambda_2 \in \mathbb{R}^m; -u_0 \preceq \lambda_1 - \lambda_2 \preceq u_0\}$ , alors la zone morte  $\phi(\lambda_1)$  vérifie l'inégalité :

$$\phi(\lambda_1)^T T(\phi(\lambda_1) + \lambda_2) \leq 0 \quad (16)$$

pour n'importe quelle matrice diagonale positive  $T \in \mathbb{R}^{m \times m}$ .

#### Exemple 2 : Illustration de la condition de secteur

Dans le cas du retour d'état, avec  $\lambda_1 = Kx$ , l'application du lemme permet d'écrire que la relation  $\phi(Kx)^T T(\phi(Kx) + Gx) \leq 0$  est vérifiée pourvu que l'état du procédé  $x$  appartienne à l'ensemble polyédral  $S(K - G, u_0) = \{x \in \mathbb{R}^n; -u_0 \preceq Kx - Gx \preceq u_0\}$ , avec  $G$  une matrice de la dimension de  $K$ , qui agit comme un degré de liberté sur le problème. Lorsque  $G = K$ , la condition de secteur est vérifiée de manière globale (dans tout  $\mathbb{R}^n$ ), tel qu'illustré sur la figure 4(a). Autrement, la condition est vérifiée de manière locale, tel qu'illustré sur la figure 4(b).

Comme cela a été mentionné dans la section 2.2 dans un contexte de stabilité, on retrouve pour la condition de secteur présentée dans le lemme précédent, les notions de propriété globale et locale, dépendantes de la condition initiale.

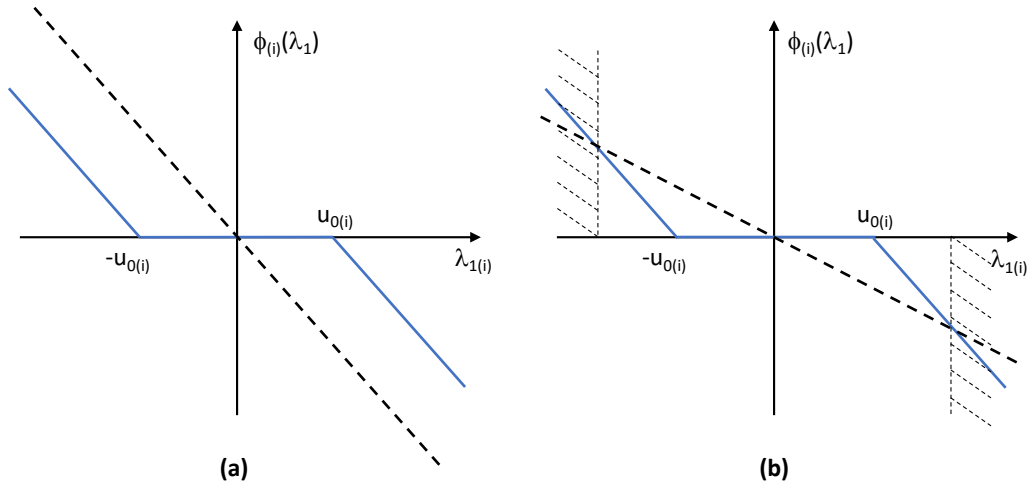


Figure 4 – Modélisation de la zone morte : (a) dans un secteur global ; (b) dans un secteur local

### 3 Architecture générale de l'anti-windup

#### 3.1 Principe

L'idée de base de la synthèse anti-windup est d'introduire une modification sur un contrôleur conçu initialement sans tenir compte des saturations de l'actionneur afin de retrouver, autant que possible, le comportement et les performances du système bouclé non saturé. Cet article est dédié à la synthèse anti-windup pour des systèmes linéaires (procédé et contrôleur) interconnectés au travers d'actionneurs présentant des saturations, selon le schéma général présenté précédemment dans la figure 1. Dans cette figure, le signal  $y_c$  produit par le contrôleur est comparé au signal contraint par l'actionneur appliqué en pratique au procédé. L'erreur entre ces deux signaux peut alors être utilisée pour modifier la stratégie de contrôle afin de mieux préserver la stabilité et les performances correspondant au système bouclé sans saturation. D'une certaine manière, cela revient à « informer » le contrôleur que ce n'est pas la valeur qu'il fournit qui est appliquée au système. Cela permet ainsi d'éviter que le contrôleur accumule des demandes de plus en plus importantes d'action, alors que le signal est saturé, en n'imaginant pas qu'elles ne sont pas réellement appliquées sur le système. Cette architecture anti-windup manipulant l'erreur  $u - y_c$  fait que le compensateur anti-windup n'est pas activé lorsque la saturation n'est pas active, ce qui garantit que, dans la zone linéaire, le contrôleur non contraint initial n'est pas modifié.

**Nota :** Il existe d'autres formes d'architectures anti-windup, telles celles basées sur des schémas de commande prédictive nommé « reference governors » qui n'entrent pas dans le cadre décrit dans cet article. Le lecteur intéressé pourra se reporter par exemple aux articles [16] ou [17].

Le schéma de principe décrit dans la figure 1 est très général et le procédé tout comme le contrôleur non contraint peuvent tout à fait être non-linéaires. Il n'existe cependant pas de méthodes génériques permettant de construire le compensateur anti-windup dans ce cas général, et cet article propose des méthodes de synthèses basées sur une description linéaire du procédé et de son contrôleur non contraint. Le compensateur anti-windup agit sur le contrôleur non contraint comme une entrée supplémentaire agissant sur la dynamique et/ou la sortie du contrôleur, ce qui conduit à modifier l'expression du contrôleur (3) de la manière suivante :

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c u_c + B_{cw} w + v_x \\ y_c &= C_c x_c + D_c u_c + D_{cw} w + v_y \end{aligned} \quad (17)$$

avec le vecteur  $\begin{bmatrix} v_x \\ v_y \end{bmatrix} \in \mathbb{R}^{n_c+m}$  correspondant à la sortie du compensateur anti-windup.

Dans son principe, le compensateur anti-windup utilise en entrée la différence entre le signal de commande contraint ( $u$ ) et non contraint ( $y_c$ ). Cela signifie que pour pouvoir être appliqué en pratique, ces deux signaux doivent être accessibles. C'est effectivement le cas lorsque l'on considère des saturations en position, le signal non contraint étant le résultat d'un calcul du bloc contrôleur et le signal contraint étant le signal physique effectivement appliqué au procédé. Dans la section 6, le cas de compensateur anti-windup venant compenser des saturations sur les dynamiques d'ordre supérieur, telles que les saturations en vitesse, sera abordé, et cet aspect lié à l'accessibilité des signaux sera discuté.

### 3.2 Architectures de compensateurs anti-windup

Différents types d'architectures anti-windup peuvent à présent être introduites, permettant de générer les signaux  $v_x$  et  $v_y$  à partir de la zone morte  $\phi(y_c)$ . La première architecture, appelée **boucle linéaire directe d'anti-windup (DLAW)**, consiste assez naturellement à construire un compensateur dynamique linéaire sous la forme :

$$\begin{cases} \dot{x}_{aw} &= A_{aw}x_{aw} + B_{aw}\phi(y_c) \\ \begin{bmatrix} v_x \\ v_y \end{bmatrix} &= C_{aw}x_{aw} + D_{aw}\phi(y_c) \end{cases} \quad (18)$$

avec l'état interne  $x_{aw} \in \mathbb{R}^{n_{aw}}$ . La synthèse du compensateur consiste dans ce cas à calculer les matrices  $A_{aw}$ ,  $B_{aw}$ ,  $C_{aw}$ ,  $D_{aw}$  de dimensions appropriées.

Le compensateur anti-windup statique est un cas particulier de cette structure pour lequel  $n_{aw} = 0$ . Il s'écrit directement sous la forme :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} D_{aw}^x \\ D_{aw}^y \end{bmatrix} \phi(y_c) = D_{aw}\phi(y_c) \quad (19)$$

Différents cas de figure peuvent ensuite être envisagés selon que l'anti-windup est prévu pour agir sur la dynamique et/ou sur la sortie du contrôleur non contraint.

Une deuxième architecture suit une stratégie différente en considérant la dynamique du compensateur anti-windup comme une copie du procédé. Cette architecture, appelée **boucle d'anti-windup basée sur une copie du modèle (MRAW)**, s'exprime sous la forme :

$$\begin{cases} \dot{x}_{aw} &= A_p x_{aw} + B_{pu}(\text{sat}(y_c) - y_c + v_y) \\ v_x &= B_c (C_p x_{aw} + D_{pu}(\text{sat}(y_c) - y_c + v_y)) \end{cases} \quad (20)$$

Dans cette stratégie, c'est le signal  $v_y$  qui doit être calculé lors de la synthèse du compensateur anti-windup. L'avantage de cette stratégie basée sur une copie du modèle est que le compensateur (20) traduit la part de la réponse du procédé manquante dans le système bouclé du fait des effets de la saturation. Indépendamment de la sélection de  $v_y$ , le système bouclé avec (2), (17) et (20) est tel que  $x_p + x_{aw}$  coïncide toujours avec la réponse du système sans contrainte, de manière à ce qu'en amenant  $x_{aw}$  à 0, on force le procédé  $x_p$  à retrouver la réponse non contrainte. Ceci permet de prouver la stabilité et les performances en utilisant les ingrédients des systèmes en cascades et a deux avantages principaux : A1) le contrôleur (17) n'a pas besoin d'être linéaire et A2) le signal  $v_y$  peut être non-linéaire. Son principal inconvénient est par contre qu'il s'agit nécessairement d'un anti-windup dynamique et que son ordre est fixé (égal à celui du système). Des détails sur cette stratégie sont donnés dans [10] et [9].

### 3.3 Formulation des problèmes

Quelle que soit l'architecture envisagée pour le compensateur anti-windup, l'objectif de la synthèse est de traiter de problèmes de stabilité et/ou de performance. Différents types de problèmes de synthèse anti-windup peuvent ainsi être définis :

PB1 Stabilité

PB1-A — pour  $w = 0$ , maximiser la région de stabilité asymptotique ;

PB1-B — pour  $w \in \mathcal{W}$ , maximiser le domaine de conditions initiales admissibles pour lesquelles les trajectoires restent bornées.

PB2 Performance

PB2-A — maximiser l'ensemble de perturbations  $\mathcal{W}$  pour lesquelles les trajectoires restent bornées ;

PB2-B — minimiser l'effet de la perturbation  $w$  sur la sortie de performance  $z$ .

Par la suite, les ensembles d'état admissibles (région de stabilité asymptotique ou de conditions initiales) sont considérés de forme ellipsoïdale, définis sous la forme :

$$\mathcal{X}_0 = \mathcal{E}(Q^{-1}, \delta) = \{x \in \mathbb{R}^n; x^\top Q^{-1} x \leq \delta^{-1}\} \quad (21)$$

avec  $\delta > 0$  et  $n$  la dimension de l'état augmenté du système bouclé, incluant les dynamiques du procédé, du contrôleur, des actionneurs et du compensateur anti-windup. De même, les ensembles de perturbations  $\mathcal{W}$  sont tels que définis dans l'équation (4), avec  $R = I_q$ . Enfin, un critère représentant l'effet de la perturbation sur la sortie  $z$  est donné par le gain  $\mathcal{L}_2$  entre  $w$  et  $z$ .

L'optimisation des régions  $\mathcal{X}_0$  peut se faire

- en maximisant la trace de  $Q$ ,
- en minimisant le trace de  $Q^{-1}$  :

$$\min \text{trace}(M_q) \quad (22)$$

$$\text{sous les contraintes } \begin{bmatrix} M_q & I_n \\ I_n & Q \end{bmatrix} > 0; \quad M_q = M_q^\top > 0, \quad (23)$$

- en minimisant un scalaire  $\beta$  tel que  $\frac{1}{\beta^2} \mathcal{E}_0 \subset \mathcal{X}_0$ , avec  $\mathcal{E}_0$  un ensemble préformé (ou défini à partir de directions privilégiées) défini par ses  $n_r$  sommets  $v_r$ ,  $r = 1 \dots n_r$  :

$$\min \beta \quad (24)$$

$$\text{sous les contraintes } \begin{bmatrix} \beta & v_r^\top \\ v_r & Q \end{bmatrix} > 0, \quad r = 1 \dots n_r \quad (25)$$

De manière similaire, l'optimisation de l'ensemble de perturbation  $\mathcal{W}$  se fait en minimisant  $\delta$ , et pour le cas où le niveau de perturbation  $\delta$  est fourni, l'optimisation de la performance de rejet de perturbation s'exprime au travers du gain  $\mathcal{L}_2$  entre les signaux  $w$  et  $z$ , donné par le scalaire  $\sqrt{\gamma}$  tel que :

$$\int_0^\infty z(t)^\top z(t) dt \leq \gamma \int_0^\infty w(t)^\top w(t) dt$$

Les problèmes ci-dessus sont définis de manière locale mais, dans la mesure où le système est asymptotiquement stable en boucle ouverte, ils peuvent dans certains cas être résolus de manière globale. Par ailleurs, dans la pratique, ces différents problèmes d'optimisation sont généralement mixés, et des critères multi-objectifs doivent être envisagés.

## 4 Réglage de correcteurs anti-windup

Dans cet article, le focus est mis sur l'anti-windup statique, décrit brièvement dans la section 3.2, et correspondant à la stratégie DLAW. Ce cas statique est en fait un cas particulier de l'anti-windup dynamique, mais présentant deux avantages majeurs :

- il est simple à mettre en oeuvre, la modification du contrôleur nominal se traduisant par un simple gain supplémentaire, sans intégration d'un modèle dynamique supplémentaire ;
- la synthèse du correcteur anti-windup s'exprime sous la forme d'inégalités matricielles linéaires, faciles à résoudre numériquement.

Enfin, du fait de sa simplicité structurelle, un réglage heuristique peut être envisagé dans des cas simples, comme cela sera discuté dans la section 4.2.

### 4.1 Réglage d'anti-windup statique

A partir de l'expression du compensateur DLAW, rappelée ci-dessous :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} D_{aw}^x \\ D_{aw}^y \end{bmatrix} \phi(y_c) = D_{aw} \phi(y_c) \quad (26)$$

le système bouclé s'écrit sous la forme générique suivante.

#### Forme générique du système bouclé avec anti-windup DLAW statique

$$\begin{aligned} \dot{x}_{bf} &= \mathbb{A}x_{bf} + (\mathbb{B}_\phi + B_v D_{aw})\phi(y_c) + \mathbb{B}_w w \\ y_c &= \mathbb{C}_y x_{bf} + (\mathbb{D}_{y\phi} + D_{vy} D_{aw})\phi(y_c) + \mathbb{D}_{yw} w \\ z &= \mathbb{C}_z x_{bf} + (\mathbb{D}_{z\phi} + D_{vz} D_{aw})\phi(y_c) + \mathbb{D}_{zw} w \end{aligned} \quad (27)$$

avec

$$\begin{aligned} B_v &= \begin{bmatrix} B_{pu} \Delta^{-1} \begin{bmatrix} 0 & I_m \end{bmatrix} \\ B_c D_{pu} \Delta^{-1} \begin{bmatrix} 0 & I_m \end{bmatrix} + \begin{bmatrix} I_{n_c} & 0 \end{bmatrix} \end{bmatrix} \\ D_{vy} &= \Delta^{-1} \begin{bmatrix} 0 & I_m \end{bmatrix} \\ D_{vz} &= D_{zu} \Delta^{-1} \begin{bmatrix} 0 & I_m \end{bmatrix} \end{aligned}$$

et les autres matrices détaillées dans (10). En considérant une fonction de Lyapunov quadratique  $V(x_{bf}) = x_{bf}^\top Q^{-1} x_{bf}$  et la condition de secteur généralisée (16), on peut construire l'algorithme suivant (solution locale) :

#### Algorithme de synthèse DLAW statique

Etape 1 : trouver une matrice  $Q$  symétrique définie positive, une matrice diagonale  $S$  définie positive, des matrices  $Z$  et  $E$  de dimensions appropriées et un scalaire  $\gamma$ , solutions du problème d'optimisation :

$$\min C_s(\mathcal{X}_0, \mathcal{W}, \gamma) \quad (28)$$

$$\begin{bmatrix} QA^\top + AQ & \mathbb{B}_\phi S + B_v E - QC_y^\top - Z^\top & \mathbb{B}_w & QC_z^\top \\ * & -2S - \mathbb{D}_{y\phi} S - S\mathbb{D}_{y\phi}^\top - D_{vy} E - E^\top D_{vy}^\top & -\mathbb{D}_{yw} & S\mathbb{D}_{z\phi}^\top + E^\top D_{vz}^\top \\ * & * & -I_q & \mathbb{D}_{zw}^\top \\ * & * & * & -\gamma I_l \end{bmatrix} < 0 \quad (29)$$

$$\begin{bmatrix} Q & Z_{(i)}^\top \\ * & \delta u_{0(i)}^2 \end{bmatrix} \geq 0, i = 1, \dots, m \quad (30)$$

Etape 2 : Extraire le compensateur anti-windup statique  $D_{aw} = ES^{-1}$ .

Quand  $w = 0$ , l'ellipsoïde  $\mathcal{E}(Q^{-1}, \delta)$  est une région de stabilité asymptotique pour le système saturé (27), et, quand  $w \neq 0$ , appartenant à l'ensemble  $\mathcal{W}$  (avec  $R = I_q$ ), les trajectoires issues de  $x_{bf}(0) = 0$  restent bornées dans  $\mathcal{E}(Q^{-1}, \delta)$  avec le gain  $\mathcal{L}_2$  entre  $w$  et  $z$  égal à  $\sqrt{\gamma}$ . Le critère d'optimisation  $\mathcal{C}_s(\mathcal{X}_0, \mathcal{W}, \gamma)$  est fixé, de même que les conditions supplémentaires associées, comme cela a été défini dans la section 3.3.

Le problème de synthèse DLAW statique peut être directement étendu au cas global en fixant  $Z = 0$ , lorsque la boucle ouverte est asymptotiquement stable (c'est-à-dire si la matrice  $A_p$  est Hurwitz. Voir aussi la section 2.2). Dans ce cas, les trajectoires restent bornées pour toute perturbation  $\mathcal{L}_2$  bornée et pour toute condition initiale  $x_{bf}(0)$ , et la norme de  $z(t)$  est telle que

$$\int_0^T z(t)^\top z(t) dt \leq \gamma \int_0^T w(t)^\top w(t) dt + \gamma x_{bf}(0)^\top Q^{-1} x_{bf}(0), \forall T \geq 0$$

**Nota :** D'un point de vue pratique, il peut être nécessaire, dans certains cas, de rajouter des conditionnements sur les variables, en particulier sur les composantes du gain anti-windup. Ceci peut se faire en rajoutant une condition de la forme :

$$\begin{bmatrix} \sigma S_{(i,i)} & E_{(j,i)} \\ E_{(j,i)} & S_{(i,i)} \end{bmatrix} > 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n_c(+m)$$

avec  $\sqrt{\sigma}$  la valeur absolue maximum pour chaque composante du gain  $D_{aw}$  (ou au moins de la partie agissant sur la dynamique du contrôleur).

## 4.2 Réglage heuristique dans le cas SISO

Dans le cas simple d'un procédé mono-entrée mono-sortie (SISO en anglais) interconnecté avec un régulateur PID, la synthèse d'un compensateur anti-windup statique s'apparente au réglage d'un simple paramètre (gain scalaire). Typiquement, si on considère un anti-windup agissant sur l'intégrateur, cela revient à rajouter un quatrième paramètre à régler, en plus des classiques gain proportionnel  $K_p$  et coefficients temps intégral  $T_i$  et temps dérivatif  $T_d$  [R7416], [18]. On considère une structure classique de PID, telle que décrite dans [R7416], de la forme :

$$y_c = K_p \left( e + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right)$$

avec  $e$  le signal d'erreur entre une entrée de référence et la sortie du procédé (de dimension 1). On ajoute le terme anti-windup en entrée de l'intégrateur, pour obtenir la structure du contrôleur PID+AW donnée dans la figure 5, avec  $K_i = \frac{K_p}{T_i}$ ,  $K_d = K_p T_d$ . L'action anti-windup est fournie par le produit du gain  $D_{aw}^x$  avec le signal d'écart entre la sortie du contrôleur  $y_c$  et l'entrée  $u_p = \text{sat}(y_c)$  effectivement transmise au procédé à contrôler.

Bien sûr, ce réglage heuristique ne garantit pas *a priori* de performance sur le système, mais permet très souvent de traiter, au moins dans un premier temps, un problème de stabilité lié à la présence d'une saturation d'actionneur. Ceci sera illustré dans la section 5.2.

## 5 Illustrations

### 5.1 Exemple de synthèse d'un correcteur anti-windup

Un modèle simplifié de la dynamique d'un hélicoptère dans le plan vertical permet d'illustrer la synthèse et l'effet d'un anti-windup sur le comportement dynamique du système. Ce modèle a été décrit dans [19] par le passé et utilisé plus récemment dans [20] en tenant compte des saturations des actionneurs (en position et vitesse). Il comporte quatre variables d'état relatives à la vitesse

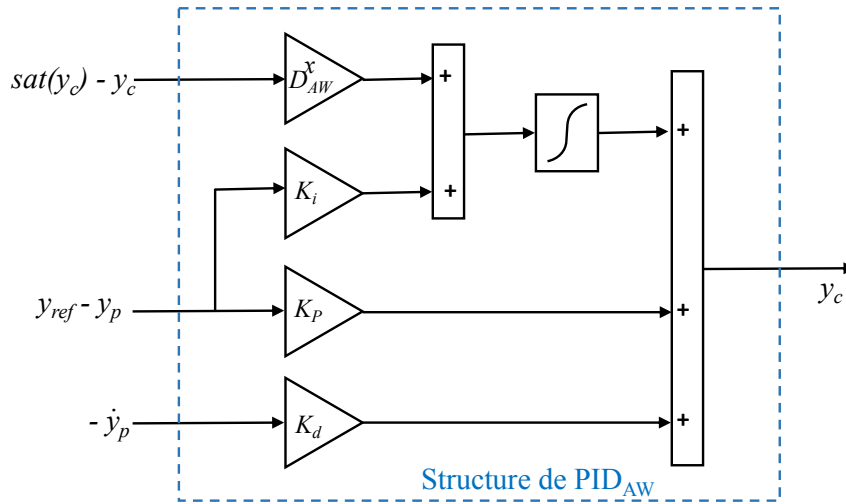


Figure 5 – Structure d'un PID avec action anti-windup

horizontale, à la vitesse verticale, à la vitesse d'inclinaison et à l'angle de tangage (ou d'inclinaison) et deux entrées de commandes : la commande de pas collectif et la commande cyclique longitudinale. La dynamique est considérée de manière approximative, comme invariante dans le temps, et l'effet des variations de vitesse sur la dynamique de la vitesse d'inclinaison est traduite par le biais d'une perturbation additive. Le modèle numérique ainsi obtenu, instable, est donné par :

$$A_p = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.0100 & 0.0024 & -4.0208 \\ 0.1002 & 0.2855 & -0.7070 & 1.3230 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}, B_{pu} = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.0447 & -7.5922 \\ -5.5200 & 4.4900 \\ 0 & 0 \end{bmatrix}, B_{pw} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C_z = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

Un contrôleur par défaut est construit pour ce système à partir de la commande *place* de Matlab, utilisée pour construire un gain de retour d'état  $K$  et un gain d'observation  $L$ , de la forme :

$$K = \begin{bmatrix} -4.7374 & -0.1737 & 1.0193 & 3.7840 \\ -3.6941 & -0.0934 & 0.5445 & 2.2728 \end{bmatrix}, L^T = \begin{bmatrix} 349.3546 & -36.4590 & 8.2464 & 1.0000 \\ -0.4555 & -4.0208 & 1.3230 & 3.0000 \end{bmatrix}$$

permettant de former les matrices du contrôleur dynamique sous la forme :  $A_c = A_p + B_{pu}K - LC_p$ ,  $B_{cu} = L$  et  $C_c = K$ . Les matrices des systèmes (2) et (3) non explicitement définies ci-dessus sont des matrices nulles de dimension appropriée.

**Nota :** L'analyse et la synthèse des correcteurs anti-windup est obtenue en utilisant les fonctions *psatal.m* et *psats1.m* de la boîte à outil SATAW-Tool disponible à l'adresse : <http://homepages.laas.fr/queinnec/sataw-tool.html>.

Dans un premier temps, le système est considéré sans perturbation ( $\delta = 0$ ) et le problème de synthèse anti-windup consiste à optimiser la région de stabilité asymptotique  $\mathcal{X}_0$  en maximisant la matrice  $Q$ . Ce problème est résolu en construisant un anti-windup statique n'agissant que sur la dynamique du contrôleur (en imposant donc  $D_{aw}^y = 0$ ). On obtient dans ce cas :

$$D_{aw}^x = \begin{bmatrix} -8.3050 & -6.1120 \\ 4.6425 & -6.9065 \\ -5.5608 & 4.4622 \\ -8.8929 & -4.2391 \end{bmatrix} \quad (31)$$



La figure 6 illustre l'effet sur la taille du domaine de stabilité de l'ajout d'un simple anti-windup statique appliqué sur la dynamique du contrôleur (la figure présente la projection de l'ensemble dans les quatre états du système hélicoptère, pour un état du contrôleur égal à zéro). La figure 7 illustre la réponse du système en boucle fermée, sans anti-windup (à gauche) et avec anti-windup (à droite), pour le système initialisé en  $x_p(0) = [0 \ 2.9 \ 0 \ -2.9]^T$ ,  $x_c(0) = 0$ . Cette condition initiale appartient au domaine de stabilité obtenu avec le correcteur anti-windup, et n'appartient pas au domaine de stabilité du système sans anti-windup.

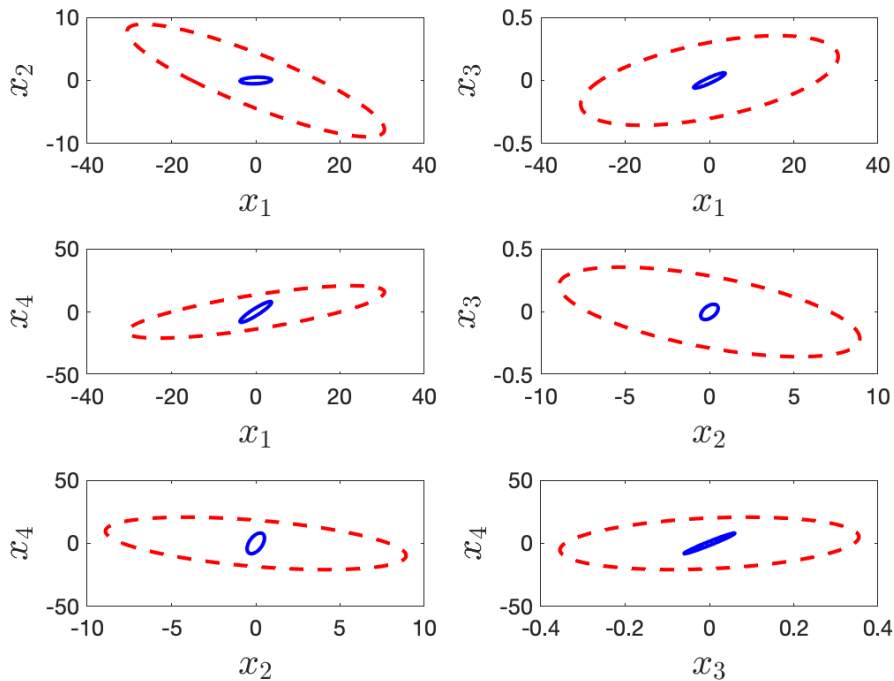


Figure 6 – Illustration : domaines de stabilité pour le système bouclé sans (trait continu bleus) et avec (tirets rouges) correcteur anti-windup.

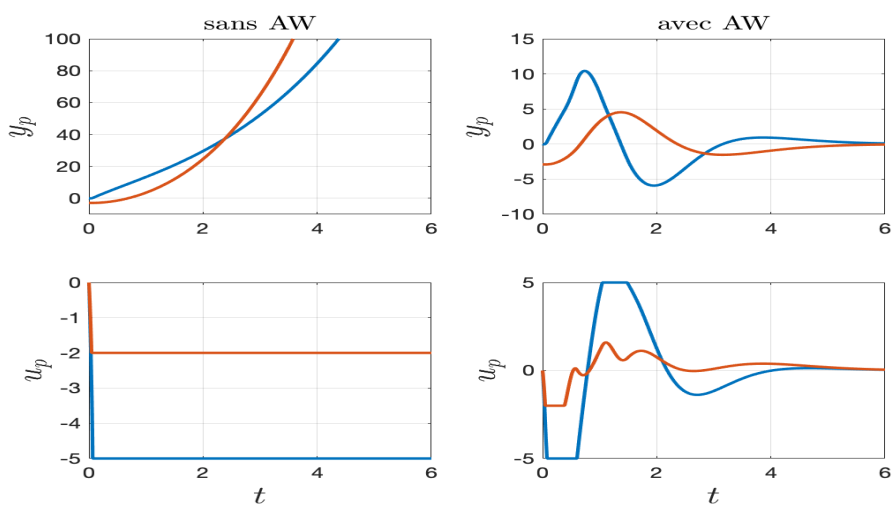


Figure 7 – Illustration : Trajectoires du système bouclé sans (figures de gauche) et avec (figures de droite) correcteur anti-windup. Les figures du haut représentent la sortie  $y_p$ , celles du bas l'entrée saturée  $u_p$ , pour le système initialisé en  $x_p(0) = [0 \ 2.9 \ 0 \ -2.9]^T$ ,  $x_c(0) = 0$ .

Le système est à présent soumis à une perturbation dont la borne est donnée par  $\delta = 20$ . C'est par exemple le cas si l'on applique un échelon unitaire pendant 0.05 secondes. Les conditions données dans l'algorithme peuvent être utilisées pour la synthèse, telles qu'elles sont présentées, ou pour l'analyse, en remplaçant dans la condition (29) la matrice  $E$  par le produit  $D_{aw}S$ , avec  $D_{aw}$  donné. Dans le cas sans anti-windup ( $D_{aw} = 0$ ), le problème de minimisation du gain  $\mathcal{L}_2$  n'est pas faisable.

A l'inverse, il est possible de vérifier que pour le correcteur  $D_{aw} = \begin{bmatrix} D_{aw}^x \\ 0 \end{bmatrix}$ , avec  $D_{aw}^x$  donné dans (31), le problème est faisable et le gain  $\mathcal{L}_2$  est donné par :  $\sqrt{\gamma} = 15.7892$ . Il est possible aussi de faire la synthèse d'un nouveau gain anti-windup, mais sans que cela change fondamentalement le résultat. La figure 8 illustre la réponse du système en boucle fermée, sans anti-windup (à gauche) et avec anti-windup (à droite), pour le système, initialement au repos, subissant une perturbation unitaire pendant 0.05s à  $t = 1$ . Comme dans le cas précédent, le système sans anti-windup diverge, tandis que les trajectoires du système avec anti-windup restent bornées et convergent vers l'origine lorsque la perturbation devient nulle.

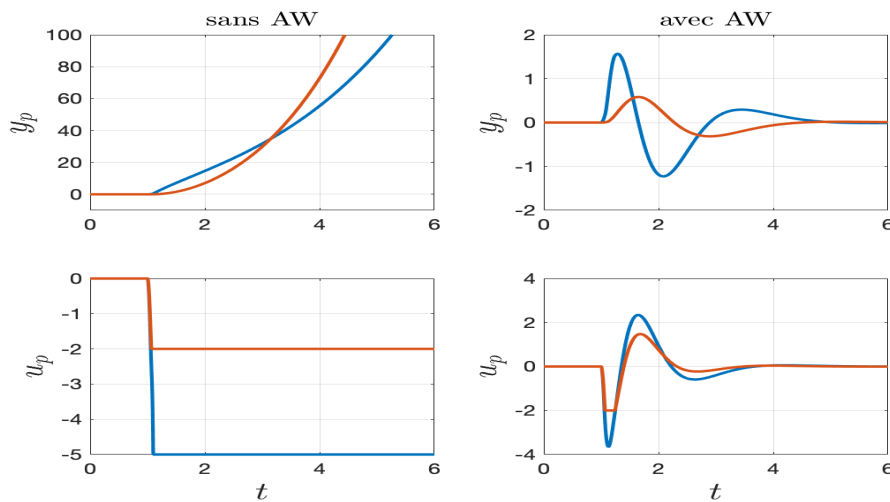


Figure 8 – Illustration : Trajectoires du système bouclé sans (figures de gauche) et avec (figures de droite) correcteur anti-windup. Les figures du haut représentent la sortie  $y_p$ , celles du bas l'entrée saturée  $u_p$  pour le système, initialement au repos, subissant une perturbation unitaire pendant 0.05s à  $t = 1$ .

## 5.2 Evaluation expérimentale d'un anti-windup statique pour un système monovarié

L'objectif de cet exemple est d'illustrer le réglage empirique d'un correcteur anti-windup sur un démonstrateur réel. Il s'agit du démonstrateur hélicoptère de Quanser [21] pour lequel un contrôleur de type PID en cascade est déjà opérationnel. L'hélicoptère est commandé en élévation par un PID reliant la mesure (et sa dérivée) de l'angle d'élévation à la commande  $u_p$  correspondant à la somme des forces appliquées sur les moteurs. L'hélicoptère est aussi commandé en voyage par deux PID en cascade. Le premier PID utilise en entrée l'erreur sur l'angle de voyage et renvoie en sortie une référence pour l'angle de tangage (angle formé entre le plan formé par la plate-forme supportant les deux hélices et le plan horizontal) qui est utilisée en entrée du second PID. Le schéma simplifié de l'installation est illustré sur la figure 9, dans laquelle est mise en évidence la saturation sur laquelle on va appliquer une correction anti-windup.

En utilisant la notation de la figure 5, le contrôleur PID+AW de voyage est donné par :

$$K_p = 3.5, \quad K_i = 1, \quad K_d = 5, \quad D_{aw}^x = -1.25$$

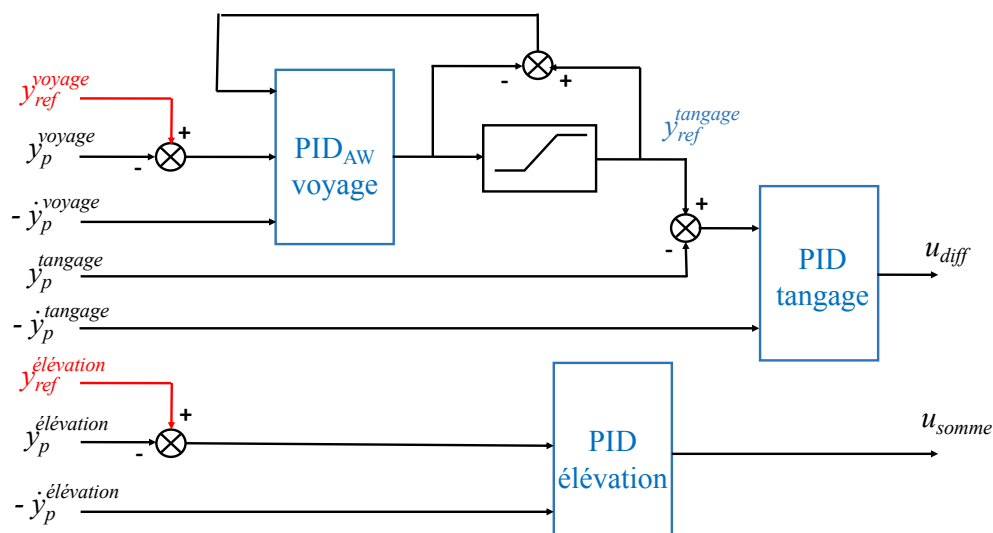


Figure 9 – Structure du contrôleur PID en cascade avec action anti-windup

Une expérience est réalisée en utilisant l'itinéraire suivant : pour l'hélicoptère initialement au repos (angles égaux à 0), à  $t = 0$ , la consigne d'élévation est fixée à 8.5 degrés. A  $t = 10$  secondes, la consigne sur le voyage est fixée à -90 degrés, qui amène l'hélicoptère à proximité d'un mur provoquant des turbulences importantes en interférant avec le déplacement d'air provoqué par les pâles de l'hélicoptère. A  $t = 50$ , la consigne de voyage est remise à 0, ce qui ramène l'hélicoptère dans une zone moins perturbée. Puis, à  $t = 80$ , un coup est donné manuellement sur la plate-forme supportant l'une des hélices afin d'engendrer une perturbation. L'expérimentation est arrêtée ensuite au bout de 2 minutes.

Les figures 10 et 12 illustrent l'évolution de la référence (en tiret) et de l'angle de voyage pour les cas sans et avec anti-windup, respectivement. Les figures 11 et 13 illustrent la sortie saturée et non saturée du contrôleur de voyage, toujours pour les cas sans et avec anti-windup.

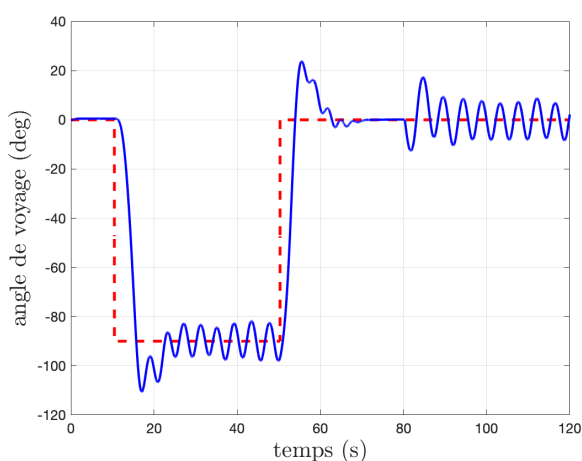


Figure 10 – Cas sans anti-windup - Evolution de la consigne en voyage et de la réponse de l'angle associée

Ces figures mettent en évidence l'efficacité de l'action anti-windup rajoutée dans le schéma de contrôle initial. Pour compléter, d'autres expériences ont été réalisées en faisant varier le gain anti-windup entre -0.75 et -2, tout en gardant la même efficacité pour ramener le système à l'équilibre. Enfin, d'autres expériences ont été réalisées pour des consignes d'élévation plus élevées (9 et 9.5), pour lesquelles la stabilité est aussi récupérée, mais au bout d'un temps d'oscillation plus long.

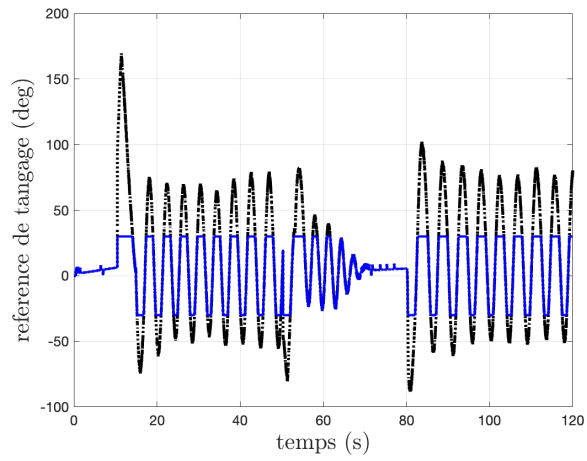


Figure 11 – Cas sans anti-windup - Evolution des sorties saturées et non saturées du contrôleur PID de voyage

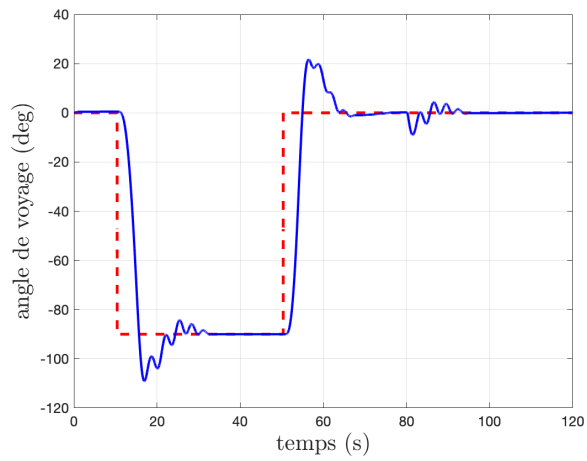


Figure 12 – Cas avec anti-windup - Evolution de la consigne en voyage et de la réponse de l'angle associée

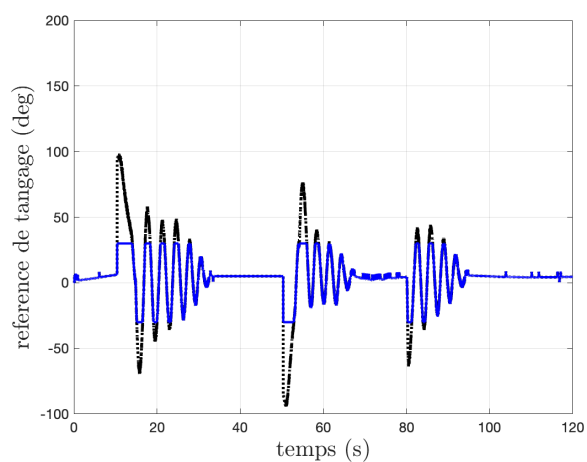


Figure 13 – Cas avec anti-windup - Evolution des sorties saturées et non saturées du contrôleur PID de voyage

Comme cela est commenté dans la section 4.2, il n'est pas possible d'établir *a priori* le domaine de fonctionnement (les trajectoires de références et/ou les perturbations autorisées), mais on peut au

moins constater que le réglage du gain anti-windup est assez facile (grande plage de gain acceptable). De même, cet exemple illustre l'extension du domaine de fonctionnement en terme de l'élévation autorisée (8 degrés au maximum pour le cas sans anti-windup, et jusqu'à 9.5 degré pour le gain anti-windup sélectionné).

## 6 Extensions

Dans cet article, nous n'avons présenté que la synthèse de contrôleur anti-windup statique. La synthèse d'anti-windup dynamiques tels que décrits dans les équations (18) et (20) suit un schéma similaire à celui proposé pour le cas statique. Les conditions théoriques et les algorithmes permettant d'obtenir les solutions numériques sont détaillés dans [9], [10].

La brique de base pour construire un compensateur anti-windup sur laquelle est focalisé l'article est la saturation en amplitude. Cependant dans la pratique (par exemple en aéronautique [6]) des actionneurs plus complexes sont à prendre en compte : ils peuvent être constitués de saturations imbriquées mêlant saturation en amplitude et en vitesse, par exemple, ou également des saturations dynamiques (dans le cas des lanceurs par exemple [22]). Le dénominateur *saturation en vitesse* regroupe plusieurs modèles dynamiques qui dépendent du phénomène physique étudié [23]. Les conditions présentées dans cet article peuvent être facilement étendues à ces cas plus complexes.

Parler de saturation en entrée ne doit pas faire oublier la présence de limitation sur la sortie, et donc la nécessité de tenir compte de capteurs non-linéaires. C'est par exemple le cas pour certains systèmes flexibles [24]. La prise en compte de saturation en sortie et la synthèse de compensateurs anti-windup est un problème compliqué car le capteur ne donne accès qu'à la fonction saturée et pas à son argument, ceci nécessitant de devoir le reconstruire. Ce problème est donc encore largement ouvert car les solutions proposées restent conservatives [25], [26].

Au-delà de l'environnement "systèmes linéaires soumis à une non-linéarité isolée" qui permet l'approximation d'une large classe de systèmes non-linéaires, il est bien évidemment pertinent de considérer la synthèse de compensateurs anti-windup pour des systèmes non-linéaires plus généraux, par exemple décrits par des équations polynomiales ou rationnelles. Il est clair alors que la technique MRAW évoquée dans le paragraphe 3.2 apparaît comme une piste intéressante. Même si des articles sont parus dans ce domaine [27], cette thématique attire beaucoup de chercheurs actuellement car ouvre sur de nouvelles architectures.

Enfin, l'approche anti-windup peut être mise en œuvre d'une manière non standard, c'est-à-dire en s'inspirant de son formalisme, mais pour aborder d'autres types de limitations, telles que, en particulier, des non-linéarités dynamiques comme le backlash [28].

## Conclusion

Cet article a abordé la synthèse d'anti-windup statique qui consiste à modifier le contrôleur préalablement synthétisé uniquement quand la saturation de l'actionneur est active. Les méthodes proposées s'appuient sur l'utilisation de conditions de secteurs afin d'encapsuler la fonction saturation et de fonctions de Lyapunov quadratiques. Ceci permet d'exprimer les conditions de stabilité et de performances par des inégalités matricielles linéaires.

Les perturbations considérées étaient de type bornées en énergie. Des algorithmes similaires pourraient être développés pour traiter des perturbations bornées en amplitude. Par ailleurs cet article

considère le cas de la régulation (stabilité de l'origine) et il est clair que le problème de poursuite est fondamental dans beaucoup d'applications. Les perturbations bornées en amplitude peuvent ainsi être utilisées pour représenter des changements de consigne. Plus généralement, des techniques de synthèse anti-windup sont déjà disponibles dans ce contexte dans la littérature, mais mériteraient d'être améliorées et unifiées afin d'avoir un meilleur impact sur les applications.

Enfin l'approche proposée considère le cas où la saturation est symétrique (voir la section 2). Les évolutions nécessaires dans le futur devront être capables de proposer des conditions non conservatives, et les plus génériques possibles, dans le cas de saturations non symétriques (contrastant avec celles disponibles dans la littérature actuellement).

## Sigles, notations et symboles

Symbole	Description
$\mathbb{R}^n$	ensemble des vecteurs réels de dimension $n$
$\mathbb{R}^{n \times m}$	ensemble des matrices réelles de dimension $n \times m$
$x^\top, A^\top$	transposée du vecteur $x$ , de la matrice $A$
$\text{He}(A)$	$A + A^\top$ (avec $A$ matrice réelle de dimension $n \times n$ )
$A > 0$ (resp. $< 0$ )	matrice $A$ symétrique et définie positive (resp. négative)
$\det(A)$	déterminant de la matrice $A$ (avec $A$ matrice réelle de dimension $n \times n$ )
$I_m$	matrice identité de dimension $m$ (l'indice est parfois omis pour simplifier les notations)
$\star$	permet de décrire les blocs symétriques de manière concise
$x_{(i)}$	$i^{\text{ième}}$ élément du vecteur $x \in \mathbb{R}^n$ , $i = 1, \dots, n$
$x \preceq y$	inégalité élément par élément, c'est-à-dire, pour 2 vecteurs de $\mathbb{R}^n$ , $x_{(i)} \leq y_{(i)}$ , $i = 1, \dots, n$
$\dot{x}$	pour un vecteur, dérivée en fonction du temps. Même notation pour une fonction

Pour en savoir plus

# Un tour d'horizon sur les techniques anti-windup pour les systèmes saturés

La commande anti-windup

par **Sophie TARBOURIECH**

Directrice de Recherche

LAAS, CNRS, Toulouse, France

par **Isabelle QUEINNEC**

Directrice de Recherche

LAAS, CNRS, Toulouse, France

## Sources bibliographiques

- [1] A.H. GLATTFELDER et W. SCHAUFELBERGER. *Control systems with input and output constraints*. Springer-Verlag, London, 2003.
- [2] P. HIPPE. *Windup in control. Its effects and their prevention*. Germany : AIC, Springer, 2006.
- [3] O. BRIEGER et al. "Anti-windup compensation of rate saturation in an experimental aircraft". In : *American Control Conference*. New York, 2007, p. 924-929.
- [4] C. ROOS et al. "On-ground aircraft control design using a parameter-varying anti-windup approach". In : *Aerospace Science and Technology* 14.7 (2010), p. 459-471.
- [5] G. FASANO et J.D. Pinter (Eds.) *Modeling and Optimization in Space Engineering*. T. 73. Springer Optimization et its Applications, 2013.
- [6] S. TARBOURIECH et al. *Pilot-Induced-Oscillations alleviation through anti-windup based approach*. T. VOL TITLE, G. Fasano and J. D. Pinter (eds). Springer, 2016.
- [7] S. TARBOURIECH et M.C. TURNER. "Anti-windup design : an overview of some recent advances and open problems". In : *IET Control Theory and Application* 3.1 (2009), p. 1-19.
- [8] S. GALEANI et al. "A tutorial on modern anti-windup design". In : *European Journal of Control* 15.3-4 (2009), p. 418-440.
- [9] S. TARBOURIECH et al. *Stability and stabilization of linear systems with saturating actuators*. Springer-Verlag, London, 2011.
- [10] L. ZACCARIAN et A.R. TEEL. *Modern anti-windup synthesis*. Princeton University Press, 2011.
- [11] T. KAILATH. *Linear systems*. Englewood Cliffs, NJ : Prentice Hall, 1980.
- [12] J. BOADA et al. "Multi-saturation Anti-windup Structure for Satellite Control". In : *American Control Conference*. Baltimore, 2010, p. 5979-5984.
- [13] L. ZACCARIAN. "Dynamic allocation for input redundant control systems". In : *Automatica* 45 (2009), p. 1431-1438.

- [14] H.J. SUSSMANN, E.D. SONTAG et Y. YANG. "A general result on the stabilization of linear systems using bounded controls". In : *IEEE Transactions on Automatic Control* 39.12 (1994), p. 2411-2425.
- [15] H.K. KHALIL. *Nonlinear systems*. MacMillan, 1992.
- [16] D. ANGELI et E. MOSCA. "Command governors for constrained nonlinear systems". In : *IEEE Transactions on Automatic Control* 44.4 (1999), p. 816-820.
- [17] E.G. GILBERT et I. KOLMANOVSKY. "Fast reference governors for systems with state and control constraints and disturbance inputs". In : *International Journal of Robust and Non-linear Control* 9.15 (1999), p. 1117-1141.
- [18] K.J. ÅSTRÖM et T. HÄGGLUND. *Automatic Tuning of PID Controllers*. Research Triangle Park : Instrument Society of America, 1988.
- [19] SCHMITENDORF. "Designing stabilizing controllers for uncertain systems using the riccati equation approach". In : *IEEE Transactions on Automatic Control* 33.4 (1988), p. 376-379.
- [20] J.M. GOMES DA SILVA, JR., S. TARBOURIECH et G. GARCIA. "Local stabilization of linear systems under amplitude and rate saturating actuators". In : *IEEE Transactions on Automatic Control* 48.5 (2003), p. 842-847.
- [21] QUANSER. *3DOF Helicopter Manual*. Available at [http://www.lehigh.edu/~inconsy/lab/frames/experiments/QUANSER-3DOFHelicopter\\_Reference\\_Manual.pdf](http://www.lehigh.edu/~inconsy/lab/frames/experiments/QUANSER-3DOFHelicopter_Reference_Manual.pdf).
- [22] S. TARBOURIECH, I. QUEINNEC et G. GARCIA. "Stability region enlargement through anti-windup strategy for linear systems with dynamics restricted actuator". In : *International Journal on Systems Science* 37.2 (2006), p. 79-90.
- [23] S.L. GATLEY et al. "A comparison of rate-limit compensation schemes for pilot-induced-oscillation avoidance". In : *Aerospace Science and Technology* 10.1 (2006), p. 37-47.
- [24] S. TARBOURIECH, C. PRIEUR et J.M. GOMES DA SILVA JR. "An anti-windup strategy for a flexible cantilever beam". In : *IFAC World Congress*. Prague, CZ, 2005.
- [25] M.C. TURNER et S. TARBOURIECH. "Anti-windup compensation for systems with sensor saturation : a study of architecture and structure". In : *International Journal of Control* 82.7 (2009), p. 1253-1266.
- [26] M. SASSANO et L. ZACCARIAN. "Model recovery anti-windup for output saturated SISO linear closed loops". In : *Systems and Control Letters* 85 (2015), p. 109-117.
- [27] L. BURLION et al. "Discontinuous model recovery anti-windup for image based visual servoing". In : *Automatica* 104 (2019), p. 41-47.
- [28] S. TARBOURIECH, I. QUEINNEC et C. PRIEUR. "Nonstandard use of anti-windup loop for systems with input backlash". In : *IFAC Journal of Systems and Control* 6 (2018), p. 33-42.

### À lire également dans nos bases

FOSSARD André J. et ORTIZ Frank – Saturation en vitesse et en vitesse-position. [S7192], Automatique avancée (2002).

LEVINE Jean et ROUCHON Pierre – Systèmes dynamiques et commande. [S7430], Régulation et commande des systèmes asservis (2008, relu et validé en 2018).

BESANCON-VODA Alina et GENTIL Sylviane – Régulateurs PID analogiques et numériques [R7416], Régulation et commande des systèmes asservis (1999).



## Outils logiciels

SATAW-Tool, a SATuration AWare matlab Toolbox, CNRS, LAAS, Toulouse, France (2013) (<http://homepages.laas.fr/queinnec/sataw-tool.html>).