Presentation of *AWAST* : the Anti-Windup Analysis and Synthesis Toolbox

J-M. Biannic and C. Roos*
(joint work with S. Tarbouriech†)

April 8, 2008

Abstract

This document presents a new MATLAB© toolbox for anti-windup analysis and synthesis. It is based on a user-friendly SIMULINK© interface which has been developed in order to simplify and fasten the definition of anti-windup design and simulation diagrams.

Keywords : Matlab toolbox, anti-windup synthesis, nonlinear systems, saturations.

1 INTRODUCTION

Anti-windup architectures were initially introduced and intuitively tuned by control engineers more than thirty years ago. Their efficiency, from a practical point of view at least, is then no longer to be proven. Theoretical works in this field, leading to more systematic anti-windup design techniques, are however much more recent. Among the first contributions which appeared hardly more than a decade ago, let us cite [8] where a unifying framework inspired by the famous standard forms from robust control theory was developed. The subject now interests numerous researchers from the control community, so that new results considering extended anti-windup schemes or introducing new optimization techniques recently appeared [6, 11, 14, 9]. In these contributions, the saturations are represented by sector nonlinearities

*ONERA/DCSD - Toulouse, France `biannic@onera.fr`
†LAAS-CNRS - University of Toulouse, France

and the anti-windup control design issue is recast into a convex optimization problem under LMI constraints. As a result, once the problem is correctly set up in an appropriate format, the anti-windup gains can be tuned automatically by efficient algorithms without requiring any extensive simulations. Following a similar path, some alternative techniques using refined representations of the nonlinearities are proposed in [7, 5, 12, 1, 10, 2].

Nevertheless, despite the above theoretical advances, the anti-windup design issue – from a control engineer perspective – remains a rather difficult and time-consuming task. This can be easily explained by the lack of user-friendly tools that would help the designer to:

- define and modify the anti-windup structure,
- perform the optimization of the gains by simply calling an appropriate routine,
- test the results with simulations.

This last comment led us to the development of this freely-available MATLAB© toolbox which implements the most recent theoretical results through a collection of analysis and anti-windup synthesis routines. Moreover, as is detailed in this document, the toolbox is based on a user-friendly SIMULINK© Library which should greatly help the designer in the definition of anti-windup synthesis and simulation diagrams.

This note is organized as follows. The anti-windup problem is briefly stated in section 2 and a few recent technical results are recalled in section 3. The toolbox is then presented in section 4. An illustrative example is given in section 5.

2 Problem statement

Consider the nonlinear interconnection of figure 1, where the saturated plant $G(s)$ to be controlled, as in [4], is written in a standard LFT form. Note that the saturations which in this diagram all appear in a diagonal nonlinear static operator Ψ will typically represent magnitude but also rate limitations in the actuators. As an example, a first-order actuator with amplitude and rate constraints can be represented by the diagram of figure 2. Note on this model that the rate limit is represented by a standard saturation at the input of the integrator block, while the magnitude constraint appears through a limited-integrator which has been approximated by a feedback loop directly inspired by anti-windup structures (more details can be found in [2]).

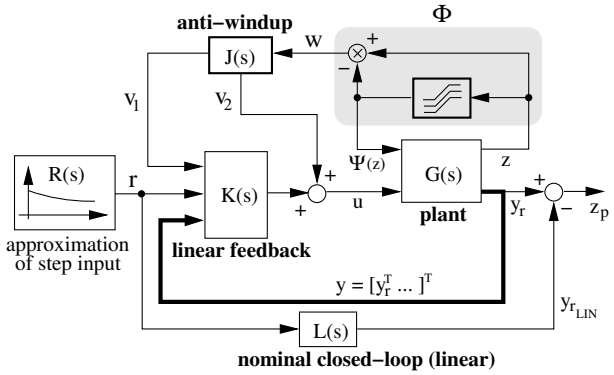

Figure 1: Standard interconnection with a general anti-windup architecture

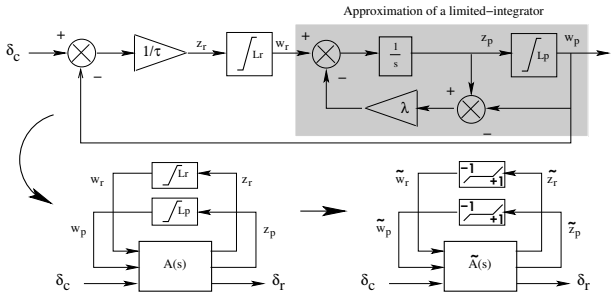

Figure 2: Representation of a saturated actuator

As is usual when considering anti-windup design problems, it is assumed that a linear controller $K(s)$ was preliminarily designed so as to stabilize the plant and ensure good performance properties in the linear region. In a second phase, following [6, 13, 14], the adverse effects of the saturations are taken into account via some additional signals v_1 and v_2 at the inputs and outputs of the nominal controller whose struture is then modified *a posteriori*:

$$K(s) : \begin{cases} \dot{x}_K = A_K\, x_K + B_K \begin{bmatrix} r \\ y \end{bmatrix} + v_1 \\ u = C_K\, x_K + D_K \begin{bmatrix} r \\ y \end{bmatrix} + v_2 \end{cases} \tag{1}$$

Such signals are obtained as the outputs of the dynamic anti-windup system $J(s)$ to be determined:

$$J(s) : \begin{cases} \dot{x}_J = A_J\, x_J + B_J\, w \ \in \mathbb{R}^{n_J} \\ v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = C_J\, x_J + D_J\, w \in \mathbb{R}^{p_J} \end{cases} \tag{2}$$

The input signal w can be interpreted as an indicator of the saturations activity. With the notation of figure 1, it is readily observed that $w = z - \Psi(z) = \Phi(z)$ where the new operator Φ still has a diagonal structure and is now composed of deadzone type nonlinearities. By rescaling the appropriate inputs and outputs of the plant $G(s)$, it can be assumed without loss of generality, that the dead-zone functions are normalized.

Finally, as is illustrated on figure 1, the exogenous input signal r is generated by a stable autonomous plant denoted $R(s)$ whose linear equations are chosen as follows (see [3]):

$$\epsilon_r \dot{r} + r = 0 \ , \quad r(0) = r_0 \in \mathbb{R}^p \tag{3}$$

As shown on the figure, this signal (approximating step inputs for small values of ϵ_r) enters both the linear controller $K(s)$ and a linear reference model $L(s)$ representing the nominal behaviour of the closed-loop plant (without saturations). The outputs of this model are then compared to the regulated nonlinear outputs which defines a performance signal z_p to be minimized. Interestingly, the introduction of the autonomous plant $R(s)$ permits to redraw (see figure 3) the diagram of figure 1 in a synthetic format without any external inputs.

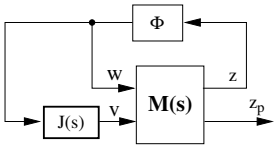

Figure 3: A synthetic view of figure 1

The linear plant $M(s)$ is easily obtained by merging the states of $R(s)$, $L(s)$, $G(s)$ and $K(s)$. Under mild assumptions on feedthrough terms, its equations read:

$$M(s) : \left\{ \begin{array}{l} \dot{\xi} = A\xi + B_\phi w + B_a v \ , \quad \xi \in \mathbb{R}^{n_M} \\ z = C_\phi \xi \in \mathbb{R}^m \\ z_p = C_p \xi \in \mathbb{R}^p \end{array} \right. \tag{4}$$

Finally, the global nonlinear closed-loop is obtained as (with $x = [\xi^T \ x_J^T]^T = [r^T \ \zeta^T]^T \in \mathbb{R}^n$):

$$\left\{ \begin{array}{l} \dot{x} = \begin{bmatrix} A & B_a C_J \\ 0 & A_J \end{bmatrix} x + \begin{bmatrix} B_\phi + B_a D_J \\ B_J \end{bmatrix} \phi(z) \\ z = [\, C_\phi \quad 0 \,]\, x \ , \quad z_p = [\, C_p \quad 0 \,]\, x \end{array} \right. \tag{5}$$

With the above notation in mind, the anti-windup synthesis problem may be stated as follows:

Problem 2.1 *With reference to figures 1 and 3, the problem is to compute a dynamic anti-windup controller $J(s)$ which, for a given amplitude $\rho = ||r_0||$ of the input signal, ensures that:*

- *the nonlinear closed-loop plant remains stable for all initial condition ζ_0 inside a domain $\mathcal{E}(\rho)$ to be computed,*
- *selected outputs of the nonlinear plant remains as close as possible to the linear reference i.e the energy of the error signal z_p is minimized.*

3 Anti-windup synthesis result

A key technical result to solve the above anti-windup synthesis problem is given below. This result is based on a recent characterization of the deadzone nonlinearity using some modified sector conditions (details can be found in [3]):

Theorem 3.1 *Consider the nonlinear closed-loop plant of figure 3. If there exist matrices:*

$$\begin{array}{l} \bullet\ Q = Q^T \in \mathbb{R}^{n\times n} \text{ with } n = n_M + n_J \\ \bullet\ S = diag(s_1, \ldots, s_m) > 0\,,\ Z = [Z_1^T \ldots Z_m^T]^T \in \mathbb{R}^{m\times n} \\ \bullet\ A_J \in \mathbb{R}^{n_J\times n_J}, B_J \in \mathbb{R}^{n_J\times m} \\ \bullet\ C_J \in \mathbb{R}^{p_J\times n_J}, D_J \in \mathbb{R}^{p_J\times m} \end{array} \tag{6}$$

and positive scalars γ and ρ such that[1]*:*

$$\begin{pmatrix} Q & \star \\ [\,\rho\, I_p \quad \mathbf{0}\,] & I_p \end{pmatrix} > 0 \tag{7}$$

$$(\star) + \begin{pmatrix} \begin{bmatrix} A & B_a\mathbf{C_J} \\ 0 & \mathbf{A_J} \end{bmatrix} Q & \begin{bmatrix} B_\phi \\ \mathbf{B_J} \end{bmatrix} S & 0 \\ S\mathbf{D_J}^T\,[\,B_a^T \quad 0\,] - Z & -S & 0 \\ [\,C_p \quad 0\,]\,Q & 0 & -\gamma I_p \end{pmatrix} < 0 \tag{8}$$

$$\begin{pmatrix} Q & \star \\ Z_i + [\,C_{\phi_i} \quad 0\,]\,Q & 1 \end{pmatrix} > 0\,,\ \ i = 1 \ldots m \tag{9}$$

then, for any bounded reference signal r defined by equation (3), with $||r_0|| \leq \rho$, the nonlinear interconnection of figure 1 is stable for all initial condition ζ_0 in the performance domain $\mathcal{E}(\rho)$ defined as follows:

$$\mathcal{E}(\rho) = \left\{ \zeta \in \mathbb{R}^{n-p} / ||r|| \leq \rho \Rightarrow \begin{bmatrix} r \\ \zeta \end{bmatrix}^T P \begin{bmatrix} r \\ \zeta \end{bmatrix} \leq 1 \right\} \tag{10}$$

[1]For compactness, the symmetric terms in the matrix inequalities are systematically replaced by "$\star$"

with $P = Q^{-1}$. Moreover, the output energy satisfies:

$$\int_0^\infty z_p(t)^T z_p(t)\, dt \leq 2\gamma \tag{11}$$

The above theorem lets appear a specifically nonlinear tradeoff between the size of the stability domain (measured by ρ) and the performance objective (evaluated by γ). In practice, this theorem can then be used either to solve a performance (by fixing ρ and minimizing γ) or a stability problem (by maximizing ρ). In both cases, the structure of the inequalities suggests the use of an LMI solver although some bilinear terms *a priori* appear in inequality (8). But interestingly, as is detailed in [1], when the matrices A_J and C_J of the anti-windup compensator are fixed, this inequality becomes linear (a simple change of variable is used for the matrices B_J and D_J: $\tilde{B}_J = B_J S$, $\tilde{D}_J = D_J S$). This means that the optimization of a dynamic anti-windup compensator with fixed poles is a convex problem. A trivial particular case is the static compensator: $J(s) = D_J$. As detailed in [1], the full-order case ($n_J = n_M$) is convex as well. In that case, the poles of the compensator are freely optimized or can also be constrained (see [10]) to avoid possibly too slow dynamics. All these possibilities are then implemented in the toolbox to be detailed in the next section.

4 Description of the toolbox

4.1 A brief overview

As illustrated by figure 4, the toolbox is composed of a root directory (*AWAST/*) which contains the two main analysis and synthesis routines (awan, awsyn) and the SIMULINK© library *AWLib*. A list of subroutines is included in the directory *sub/*. To get easily started with the toolbox, an illustrative example is given in the *demo/* directory.

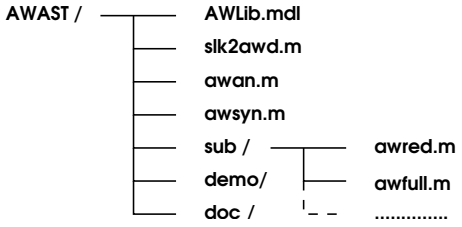

Figure 4: Organization of the toolbox

4.2 The Simulink© library: AWLib

The graphic library *AWLib* (see figure 5) is an essential component of the toolbox.

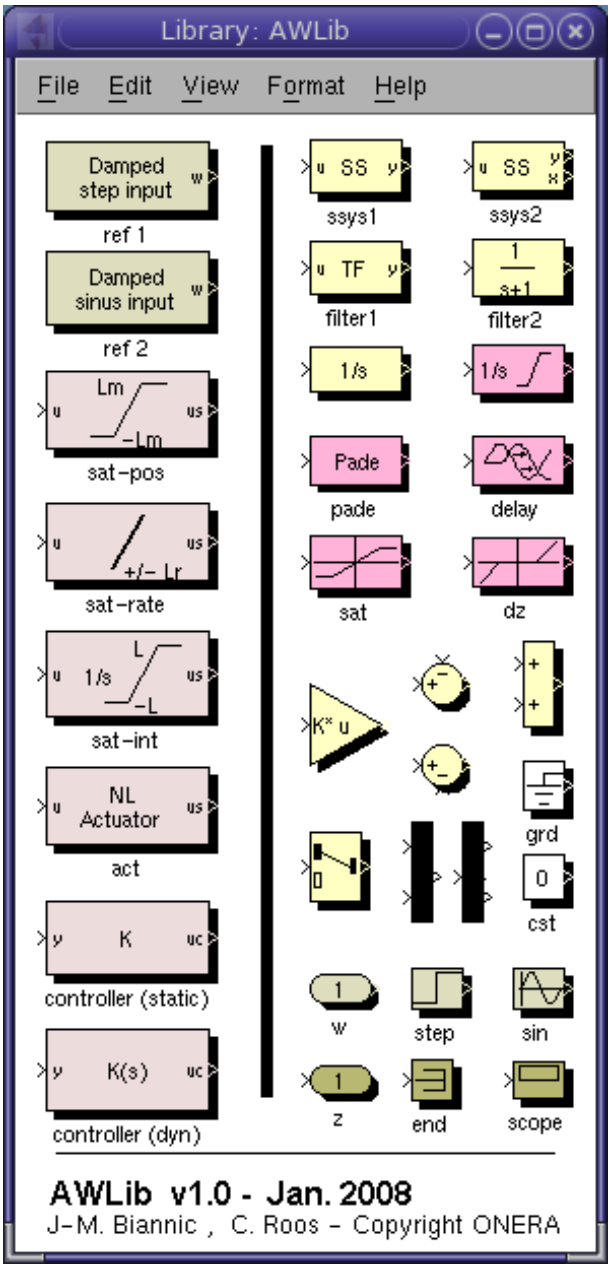

Figure 5: Simulink library

On its left part, it contains advanced SIMULINK© objects such as reference signals generators, magnitude saturations, efficient approximations of rate saturations and limited-integrators, a complete nonlinear actuator model with mixed magnitude and rate limitations, and finally linear compensators (static or dynamic) with anti-windup inputs. All such blocks have been designed in order:

- to ensure full compatibility with the analysis and design tools, by only involving static normalized deadzone nonlinearities. When required, following the ideas illustrated by figure 3, conversions and normalizations are automaticaly achieved within each block,
- to fasten the construction of SIMULINK© diagrams. Morover, according to the way it is parameterized (see below), the same block can either be used in a design or simulation oriented diagram. This permits to switch rapidly and securely from design to simulation. designed.

To make the library self-sufficient, many useful standard blocks – located on the right part of the window – have also been included.

Once they have been moved to a SIMULINK© file, all blocks are easily parameterized with the help of a dialog box. An example is given on figure 6 for the case of a second-order nonlinear actuator.

The first parameter (check-box type) which is circled in red is essential since it permits to switch from a design to a simulation-oriented block. In this last case, the colour of the block changes and additional outputs w appear to enable the connection of the anti-windup compensator. The next parameters enable the user to specify the magnitude and rate limitations, the pulsation and the damping of the actuator. The last parameter is used to sort the anti-windup signals in case of a complex diagram involving several controllers or to make the saturations non visible by the anti-windup device (in such a case, the parameter is set to 0 and a partial anti-windup will be computed).

4.3 The main synthesis routine: awsyn

Assume that a design diagram *dsg-diag.mdl* was built with the help of the above library. The anti-windup synthesis is then performed very easily thanks to the main routine of the toolbox via the following MATLAB© command line:

```
>>[J,P,crit]=awsyn('dsg-diag',opt);
```

where the optional argument *opt* may either be a complex-valued vector which permits to specify the poles of the compensator or a string (*opt* =

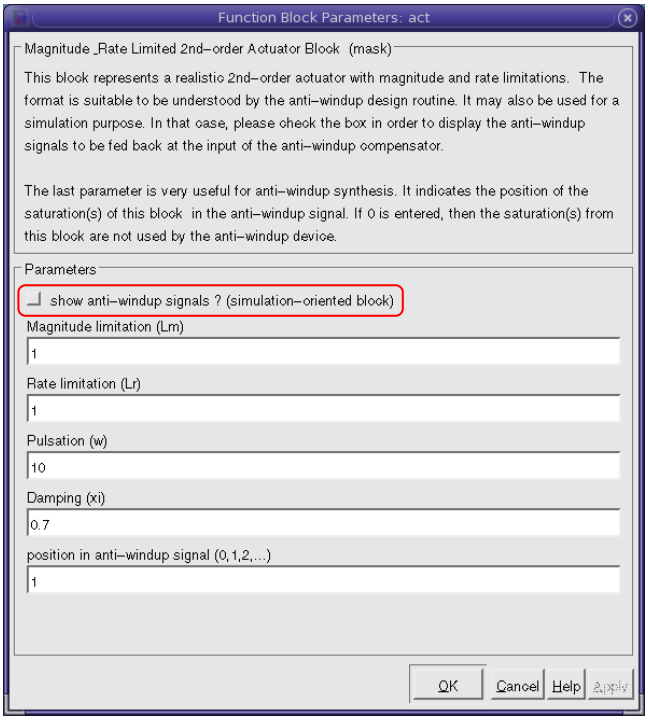

Figure 6: Dialog box of the nonlinear actuator

'full') if a full-order anti-windup controller is to be computed. By default, a static gain is optimized.

The routine starts by analyzing the design diagram and converting it into a standard plant $M(s)$ as is presented in section 2 of the paper. According to the parametrization beeing used in the *damped inputs* blocks, a stability or a performance problem will be solved (this point will be clarified in the next section). Then, according to the value of the optional parameter *opt*, the appropriate subroutine is invoked to solve either a full or fixed-order design problem. In this last case, the matrices A_J and C_J of the controller are fixed as proposed in [1] and the LMI optimization problem of Theorem 3.1 is solved via LMILAB©.

5 Illustration

A short illustration of the toolbox to a combat-aircraft longitudinal flight control problem is now presented. It is assumed that a PID control system was preliminarily designed in order to track the angle-of-attack (α) as fast as possible around a critical point in the flight envelope ($Mach = 0.3$, $H =$

5000 ft) for which the open-loop plant is **unstable**. The linearized equations of the aircraft are as follows:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.5 & 1 \\ 0.8 & -0.4 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} - \begin{bmatrix} 0.2 \\ 5 \end{bmatrix} \delta_e \tag{12}$$

where α, q and δ_e respectively denote the angle-of-attack, the pitch rate and the elevator deflection.

Although the PID control law performs well as long as the amplitudes of the commanded angles (α_c) remain small, some severe difficulties appear for larger pilot inputs ($\alpha_c > 8\,deg$). This is easily explained by the presence of rate and magnitude saturations in the actuator, which is represented by a second-order plant ($\eta = 0.6$ and $\omega = 60\,rad/s$).

In a first step, a full-order dynamic compensator is designed so as to maximize the amplitude of α_c while preserving the stability of the nonlinear closed-loop plant. To this purpose, a design diagram is built with the help of the dedicated library. As illustrated by figure 7 this diagram exhibits four main elements:

- a *damped step-input* block to generate α_c whose maximum amplitude ρ (via the first parameter of the associated dialog box) is set to a negative value -1. This means that the parameter ρ is not fixed *a priori* but has to be maximized,
- a *dynamic controller (with anti-windup entries)* block. The first parameter (check-box type) is not activated here since the block is used in a design diagram. The second parameter contains a standard state-space description of the PID nominal controller. The last parameter (popup type) is set on *inputs and outputs* which means that the two signals v_1 and v_2 as they appear in equation (1) are used,
- a *nonlinear actuator (magnitude & rate limited 2nd-order)* block whose parameters (see figure 6) are fixed according to the above specifications,
- a *linear system* block which contains a state-space description of the linearized aircraft described by equation (12).

Note finally that this first *stability-oriented* design diagram does not contain any output.

Once the diagram is built, a full-order anti-windup compensator, maximizing ρ is computed by the following command line:

```
>>[J,P,rho]=awsyn('stab-design','full');
```

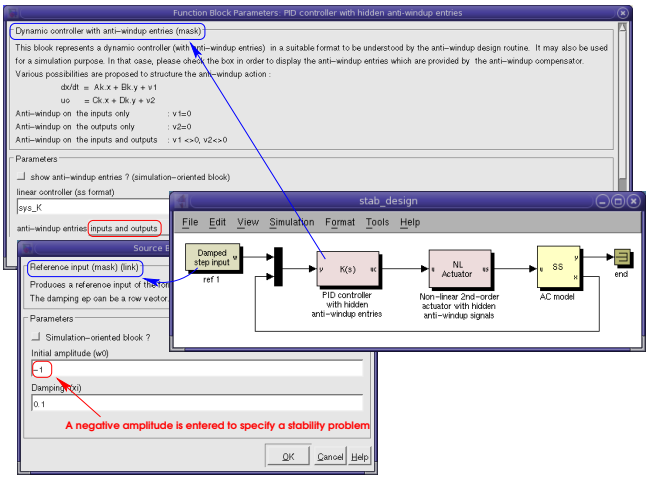

Figure 7: Stability-oriented design diagram

For this application, the optimization on a Sun Ultra 45 workstation is performed in a few seconds and a 6^{th}-order compensator is obtained. The optimized parameter ρ nearly reaches $29\,deg$, which is about four times larger than the highest admissible value without any anti-windup device. The stability domain is then considerably enlarged.

Let us now check this property with time-domain simulations. Based on the design diagram of figure 7, a simulation-oriented diagram is then derived. As illustrated on figure 8, the new diagram is readily obtained by simply modifying the properties of the controller and actuator blocks. These two blocks now respectively exhibit additional inputs and outputs enabling the connection of the anti-windup compensator. Interestingly, the construction of this new diagram is then achieved in a few seconds. Moreover the compatibility between design and simulation is automatically guaranteed. As expected, the time-domain simulations reveal that the nonlinear closed-loop plant now remains stable for large step-input signals (up to $30\,deg$).

However, as shown on figure 10, the performance is very poor. The response-time indeed exceeds $10\,sec$, while it should be less than $1\,sec$! To improve this point, a *performance-oriented* design is to be considered. The diagram of figure 7 is then updated as shown on figure 9. It now includes an output signal z_p expressing the difference between the nonlinear output α and a reference signal generated by a linear reference model. The latter, denoted $L(s)$ in section 2, is simply obtained as a second-order balanced reduction of the nominal closed-loop plant. Note finally that a fixed and positive value is now chosen for ρ in the dialog box of the *damped step input* block. This

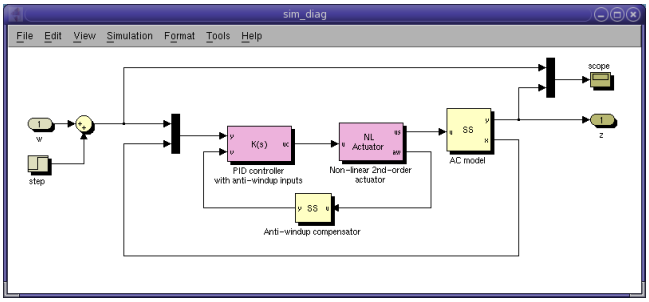

Figure 8: Simulation diagram

value corresponds to the amplitude of α_c for which the energy of the tracking error z_p is to be minimized.

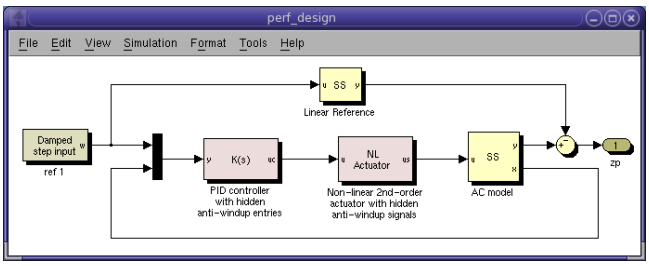

Figure 9: Performance-oriented design diagram

As for the stability case, a full-order anti-windup compensator is preliminarily designed. The parameter ρ is fixed to $8\,deg$ which corresponds to the critical value which was observed without any anti-windup device. The routine **awsyn** is then invoked as follows:

```
>>[J,P,crit]=awsyn('perf-design','full');
```

and a 8^{th}-order controller is obtained (the additional two states are introduced by the reference model). The performance index is rather small (*crit=0.33*) which means that the nonlinear output should remain close to the linear response. This is indeed confirmed by the time-domain simulation (see the magenta plot on figure 10) which moreover reveals that the response is still correct for large amplitude inputs (up to $20\,deg$). In a next step, a third-order controller is computed by selecting a few poles from the full-order solution. The selection is made by eliminating fast and slow dynamics. The reduced-order controller is then obtained as follows:

```
>>[J,P,crit]=awsyn('perf-design',[-3 -6+2*j]);
```

Interestingly, no degradration is observed on the performance index and the time-domain responses cannot even be distinguished on figure 10. Finally,

a static anti-windup gain is computed by simply invoking the synthesis function without any optional argument. In this case, the performance is slightly degraded ($crit = 0.56$) which is also visible on the simulations results. The response-time slightly exceeds 1 *sec* when the static anti-windup gain is used, while it remains below 0.9 *sec* with the dynamic compensators.

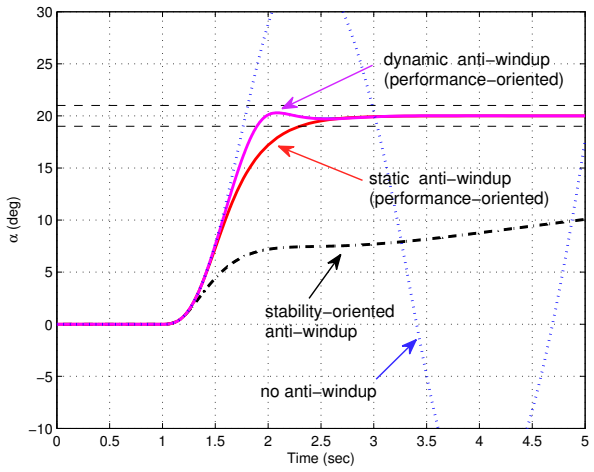

Figure 10: Simulations results

References

[1] J-M. Biannic, C. Roos, and S. Tarbouriech. A practical method for fixed-order anti-windup design. *17th IFAC Symposium on Nonlinear Control Systems*, Pretoria, August. 2007.

[2] J-M. Biannic and S. Tarbouriech. Optimization and implementation of dynamic anti-windup compensators with multiple saturations. *Submitted to Control Engineering Practice*, December. 2007.

[3] J-M. Biannic and S. Tarbouriech. Stability and performance enhancement of a fighter aircraft flight control system by a new anti-windup approach. *17th IFAC Symposium on Automatic Control in Aerospace*, Toulouse, France, June 2007.

[4] J-M. Biannic, S. Tarbouriech, and D. Farret. A practical approach to performance analysis of saturated systems with application to fighter

aircraft flight controllers. In 5^{th} *IFAC Symposium ROCOND*, Toulouse, France, July 2006.

[5] J.M. Gomes da Silva Jr. and S. Tarbouriech. Anti-windup design with guaranteed regions of stability: An LMI-based approach. *IEEE Transactions on Automatic Control*, 50(1):106–111, January 2005.

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

[7] T. Hu, A.R. Teel, and L. Zaccarian. Nonlinear $\mathcal{L}_2$ gain and regional analysis for linear systems with anti-windup compensation. In *Proceedings of the ACC*, pages 3391–3395, Portland, OR, USA, June 2005.

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

[9] B. Lu, F. Wu, and S. Kim. Linear parameter varying antiwindup compensation for enhanced flight control performance. *AIAA Journal of Guidance, Control and Dynamics*, 28(3):494–504, 2005.

[10] C. Roos and J-M. Biannic. A convex characterization of dynamically-constrained anti-windup controllers. *To appear in Automatica*, 2008.

[11] M. Saeki and N. Wada. Synthesis of a static anti-windup compensator via linear matrix inequalities. *International Journal of Robust and Nonlinear Control*, 12:927–953, 2002.

[12] S. Tarbouriech, I. Queinnec, and G. Garcia. Stability region enlargement through anti-windup strategy for linear systems with dynamics restricted actuator. *International Journal of System Science, Special issue on anti-windup*, 37(2):79–90, February 2006.

[13] F. Wu and B. Lu. Anti-windup control design for exponentially unstable LTI systems with actuator saturation. *Systems and Control Letters*, 52:305–322, 2004.

[14] F. Wu and M. Soto. Extended anti-windup control schemes for LTI and LFT systems with actuator saturations. *International Journal of Robust and Nonlinear Control*, 14:1255–1281, 2004.