Automatic Control 2

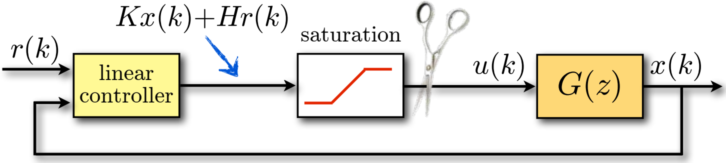
Anti-windup techniques

Prof. Alberto Bemporad

University of Trento

Academic year 2010-2011

Problem



- Most control systems are designed based on linear theory
- A linear controller is simple to implement and performance is good, as long as dynamics remain close to linear
- Nonlinear effects require care, such as *actuator saturation* (always present)
- Saturation phenomena, if neglected in the design phase, can lead to closed-loop instability, especially if the process is open-loop unstable
- Main reason: the control loop gets broken if saturation is not taken into account by the controller: $u(k) \neq Kx(k)$ for some k

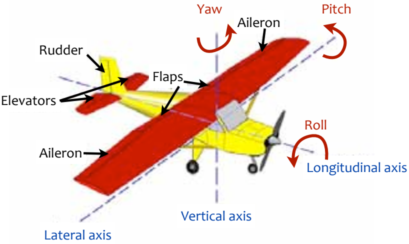
[1] K.J. Åstrom, L. Rundqwist, "Integrator windup and how to avoid it", Proc. American Control Conference, Vol. 2, pp. 1693-1698, 1989

Example: AFTI-F16 aircraft

Linearized model:

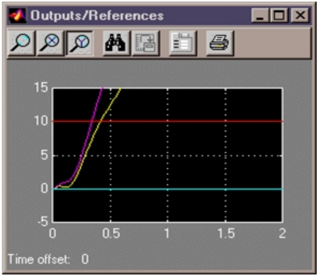
$$\begin{cases} \dot{x} = \begin{bmatrix} -0.0151 & -60.5651 & 0 & -32.174 \\ -0.0001 & -1.3411 & 0.9929 & 0 \\ 0.00018 & 43.2541 & -0.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2.516 & -13.136 \\ -0.1689 & -0.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \end{cases}$$

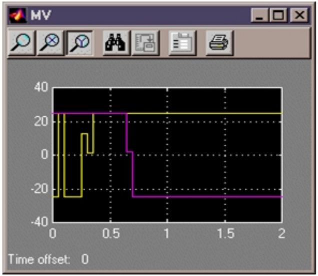
- Inputs: elevator and flaperon (=flap+aileron) angles
- Outputs: attack and pitch angles
- Sampling time: $T_s = 0.05$ sec (+ ZOH)
- Constraints: $\pm 25°$ on both angles
- Open-loop response: unstable open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$



Example: AFTI-F16 aircraft

- LQR control + actuator saturation $\pm 25°$



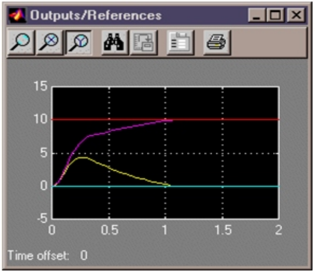


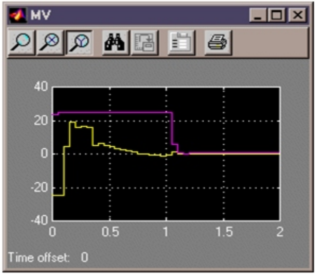
The system is unstable !

Actuator saturation cannot be neglected in the design of a good controller

Example: AFTI-F16 aircraft

- With a controller designed to handle saturation constraints:



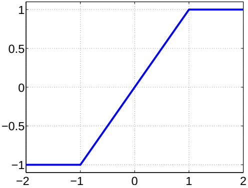


- There are several techniques to handle input saturation, the most popular ones are *anti-windup* techniques

Saturation function

- Saturation can be defined as the static nonlinearity

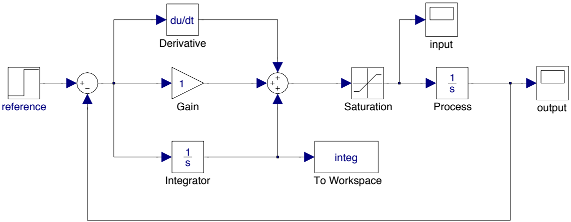
$$\text{sat}(u) = \begin{cases} u_{\text{min}} & \text{if} \quad u < u_{\text{min}} \\ u & \text{if} \quad u_{\text{min}} \le u \le u_{\text{max}} \\ u_{\text{max}} & \text{if} \quad u > u_{\text{max}} \end{cases}$$



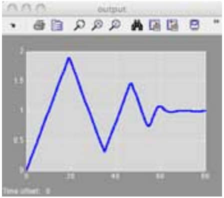
- u_{min} and u_{max} are the minimum and maximum allowed actuation signals (example: ± 12 V for a DC motor)
- If u is a vector with m components, the saturation function is defined as the saturation of all its components ($u_{\text{min}}, u_{\text{max}} \in \mathbb{R}^m$)

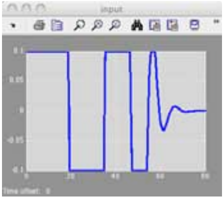
$$\text{sat}(u) = \begin{bmatrix} \text{sat}(u_1) \\ \text{sat}(u_2) \\ \vdots \\ \text{sat}(u_m) \end{bmatrix}$$

The windup problem



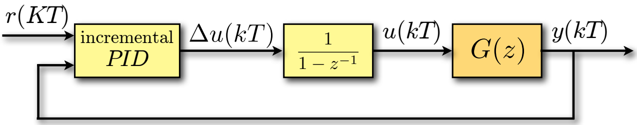
very simple process (=an integrator)
controlled by a PID controller ($K_P = K_I = K_D = 1$)
under saturation $-0.1 \leq u \leq 0.1$





- The output takes a long time to go steady-state
- The reason is the "windup" of the integrator contained in the PID controller, which keeps integrating the tracking error even if the input is saturating
- *anti-windup* schemes avoid such a windup effect

Anti-windup #1: incremental algorithm



- It only applies to PID control laws implemented in incremental form

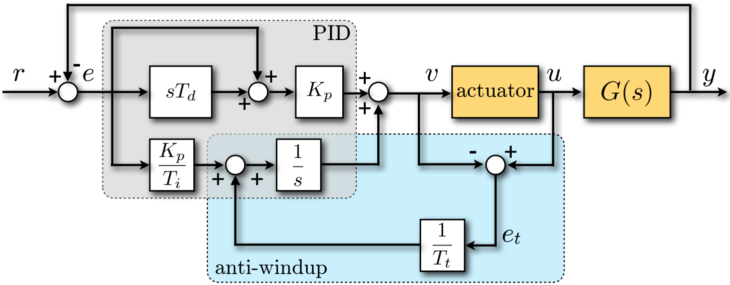
$$u((k+1)T) = u(kT) + \Delta u(kT)$$

where

$$\Delta u(kT) = u(kT) - u((k-1)T)$$

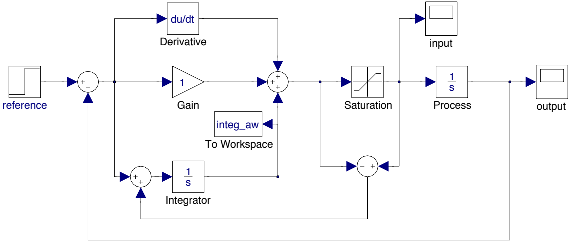
- Integration is stopped if adding a new $\Delta u(kT)$ causes a violation of the saturation bound

Anti-windup #2: Back-calculation

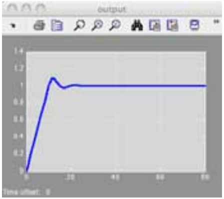


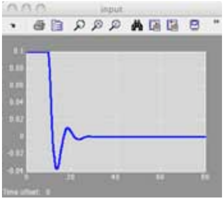
- The anti-windup scheme has no effect when the actuator is not saturating ($e_t(t) = 0$)
- The time constant T_t determines how quickly the integrator of the PID controller is reset
- If the actual output $u(t)$ of the actuator in not measurable, we can use a mathematical model of the actuator. Example: $e_t(t) = v(t) - \mathrm{sat}(v(t))$

Anti-windup #2: Back-calculation



anti-windup scheme for a PID controller ($K_P = K_I = K_D = 1$, $T_t = 1$)

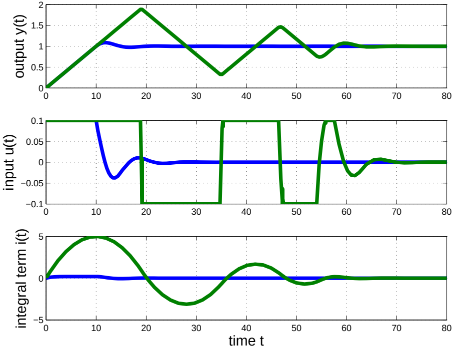




- Integrator windup is avoided thanks to back-calculation

Benefits of the anti-windup scheme

- Let's look at the difference between having and not having an anti-windup scheme



Note that in case of windup we have

- strong output oscillations
- a longer time to reach the steady-state
- the peaks of control signal

Anti-windup #3: Conditional integration

- The PID control law is

$$u(t)=K_p(br(t)-y(t))+I(t)-K_pT_d\frac{dy(t)}{dt}=K_pbr(t)-K_py_p(t)+I(t)$$

where $y_p(t)=y(t)+T_d\frac{dy(t)}{dt}$ is the prediction of the output for time $t+T_d$

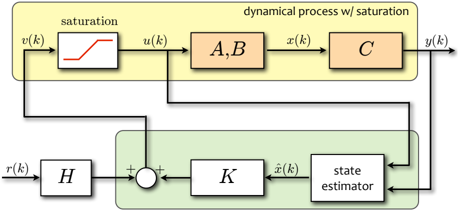
- Consider the *proportional band* $[y_l(t),y_h(t)]$ for $y_p(t)$ in which the corresponding u is not saturating

$$\begin{aligned} y_l(t) &= br(t)+\frac{I(t)-u_{\max}}{K_p} \\ y_h(t) &= br(t)+\frac{I(t)-u_{\min}}{K_p} \end{aligned}$$

where $u_{\min}$ and $u_{\max}$ are the saturation limits of the actuator

- The idea of conditional integration is to update the integral term $I(t)$ only when y_p is within the proportional band (for PI control simply set $y_p=y$)
- An hysteresis effect may be included to prevent chattering

Anti-windup #4: Observer approach



- The anti-windup method applies to dynamic compensators

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) \\ \qquad\qquad\quad + L(y(k) - C\hat{x}(k)) \\ u(k) = K\hat{x}(k) + Hr(k) \end{cases}$$

- by simply feeding the saturated input to the observer

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + B\,\text{sat}(v(k)) \\ \qquad\qquad\quad + L(y(k) - C\hat{x}(k)) \\ v(k) = K\hat{x}(k) + Hr(k) \\ u(k) = \text{sat}(v(k)) \end{cases}$$

Example

- The process to be controlled is obtained by exactly sampling

$$\begin{cases} \dot{x} = \begin{bmatrix} -1.364 & 0.4693 & 0.736 & 1.131 \\ -1.08 & -1.424 & 0.1945 & -0.7132 \\ 0.0499 & 0.8704 & -0.9675 & -0.3388 \\ -0.9333 & 0.8579 & -0.5436 & -0.9997 \end{bmatrix} x + \begin{bmatrix} 0.05574 \\ 0 \\ -0.04123 \\ -1.128 \end{bmatrix} u \\ y = \begin{bmatrix} -1.349 & 0 & 0.9535 & 0.1286 \end{bmatrix} x \end{cases}$$

  with $T = 0.5s$
- The input u saturates between ± 2
- Control design by pole placement: poles in e^{-5T}, e^{-5T}, $e^{(-2\pm j)T}$
- Observer design by pole placement: 4 poles in e^{-10T}
- The dynamic compensator is

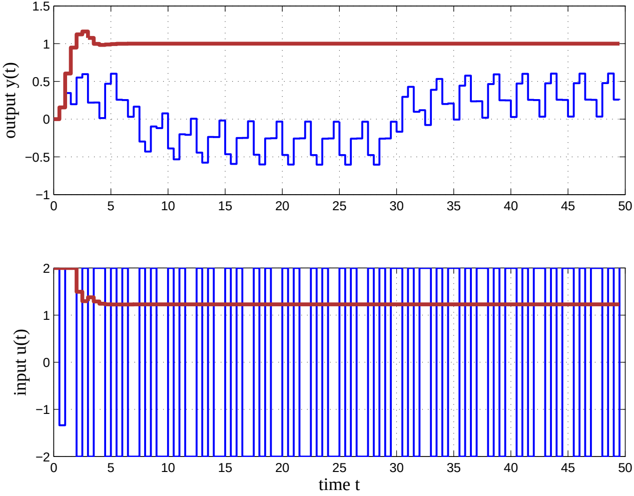
$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + B\,\text{sat}(v(k)) + L(y(k) - C\hat{x}(k)) \\ v(k) = K\hat{x}(k) + Hr(k) \end{cases}$$

  with

$$K = \begin{bmatrix} 0.4718 \\ -1.5344 \\ -2.8253 \\ 2.1819 \end{bmatrix}', \; L = \begin{bmatrix} 1.1821 \\ 1.1924 \\ 4.2054 \\ -3.6554 \end{bmatrix}, \; H = 1/(C(I-A-BK)^{-1}B) = 8.2668$$

Example (cont'd)

- Compare the results with anti-windup and without anti-windup:



Conclusions

- Conditional integration is easy to apply to many controllers, although it may not be immediate to find the conditions to block integration and to avoid chattering
- Back-calculation only requires tuning one parameter, the time constant T_t. But it only applies to PID control
- The observer approach is very general and does not require tuning any additional parameter. It also applies immediately to MIMO (multi-input multi-output) systems
- Saturation effects can be included in an optimal control formulation. In this case the control action is decided by a constrained optimization algorithm, as in *model predictive control (MPC)* techniques

English-Italian Vocabulary

	
anti-windup scheme	*schema di anti-windup*
aileron	*alettone*
pitch	*beccheggio*
roll	*rollio*
yaw	*imbardata*
model predictive control	*controllo predittivo*

Translation is obvious otherwise.