Simple Anti-Windup Controllers

Radhakisan Sohanlal Baheti
GE Research and Development Center, Schenectady, NY 12301

Abstract

The paper describes digital implementation of proportional, integral, and derivative (PID) compensators for a class of multiple-input multiple-output dynamic systems to regulate actuators in a coordinated manner. In practice, actuators of systems to be controlled are constrained to preset operating ranges. Integrator windups can occur if any of the actuators saturate and the controller continues to integrate the errors. The situation can often lead to a significant performance degradation and may result in unstable operation. A systematic approach is presented to implement a multivariable controller that is free from integrator windup. The concept is based on conditional integration and coordination to avoid windup in any of the integrators. The method is applicable to a digital multivariable controller with lead and lag actions. The concept has been demonstrated on a resaturator in the integrated gassification combined cycle (IGCC) process evaluation facility.

1 Introduction

A proportional, integral, and derivative (PID) of the error signal is the most common control action in feedback control systems. A reset windup or integrator saturation can occur if an actuator saturates and the regulator continues to integrate the error signal. The output of the integrator can then assume very large values and it can take long time to get back to a normal value. One way to avoid the windup for a single-input single-output regulator is to stop integration when the actuator is in the limit [1]. However, in a multiple-input multiple-output system, any two integrator outputs can assume large values while the corresponding actuator is still away from the limits. Further, the windup may not show in simulation studies and may surface while the system is in operation. A systematic coordination between the integrators is necessary to avoid the windup [2].

2 SISO Systems

The implementation of a single-input single-output digital PID controller to avoid the integrator saturation is discussed in reference 1. The basic principle is to use the integral part of the regulator when the error is sufficiently small and reset the integral part to a low or high value when the actuator is saturated.

A digital PI regulator for a single-input single-output dynamic system with a proportional and integral gains can be represented by [1]

$$u(t) = K\left(1 + \frac{\Delta t}{T_I(1-q^{-1})}\right)e(t) \qquad (1)$$

$$t = 0, 1, 2 \ldots$$

where $u(t)$ and $e(t)$ are the controller output and the controller input signals respectively. t is the discrete-time and Δt is the sampling interval. K is the proportional gain and T_I is the reset time constant. q^{-1} is a backward shift operator defined by

$$q^{-1}u(t) = u(t-1) \qquad (2)$$

Equation (1) can be written as

$$u(t) = u(t-1) + Ke(t) + K_I e(t) - Ke(t-1) \qquad (3)$$

where $K_I = K\Delta t/T_I$. The control signal $u(t)$ can be limited to account for saturation in the actuator. The actuator command $u_c(t)$ is given by

$$u_c(t) = \begin{cases} u_l & u(t) \le u_l \\ u(t) & u_l < u(t) < u_h \\ u_h & u(t) \ge u_h \end{cases} \qquad (4)$$

where u_l and u_h denote the lower and upper limits on the control signal $u(t)$. A block diagram of a single-input single-output PI regulator is shown in Fig. 1. The implementation is based on a backward difference approximation for the integral term in a continuous-time PI controller [1]. The reference command, system output, system input are denoted by $r(t), y(t)$ and $u_c(t)$ respectively. The delay operator provides one sample time delay.

From equations (3) and (4), the control signal $u(t)$ can be computed as

$$u(t) = u_c(t-1) + Ke(t) + K_I e(t) - Ke(t-1) \qquad (5)$$

The method can be applied to a digital PID regulator given by [1]

$$u(t) = K\left(1 + \frac{\Delta t}{T_I(1-q^{-1})} + \frac{T_D(1-q^{-1})}{\Delta t(1+\gamma q^{-1})}\right)e(t) \qquad (6)$$

where T_D is the discrete-time equivalent of a derivative gain. γ is a constant.

Equation (6) can be written as

$$u(t) = a_1 u(t-1)+a_2 u(t-2)+b_0 e(t)+b_1 e(t-1)+b_2 e(t-2) \quad (7)$$

where

$$\begin{aligned} a_1 &= (1-\gamma) \\ a_2 &= \gamma \\ b_0 &= K(1+\frac{\Delta t}{T_I}+\frac{T_D}{\Delta t}) \\ b_1 &= K(\gamma - 1 + \frac{\gamma \Delta t}{T_I} - \frac{2T_D}{\Delta t}) \\ b_2 &= K(-\gamma + \frac{T_D}{\Delta t}) \end{aligned}$$

From eequations (4) and (7), the control signal $u(t)$ can be computed as

$$u(t) = a_1 u_c(t-1)+a_2 u_c(t-2)+b_0 e(t)+b_1 e(t-1)+b_2 e(t-2) \quad (8)$$

The method is applicable to a single-input single-output digital compensator that can be represented by a higher-order difference equation.

3 MIMO Systems

A straightforward generalization of the single-input single-output anti-windup for multi-input multi-output system may lead to undesirable situations. For example, for a two-input two-output system, the two integrators with large values may cancel each other, while both the actuators may still be within limits. A systematic procedure is required where the integral terms always remain within a normal range.

Method I

A class of digital compensators with a proportional and integral gains for a two-input two-output system can be represented by

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} h_{11}(q^{-1}) & h_{12}(q^{-1}) \\ h_{21}(q^{-1}) & h_{22}(q^{-1}) \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} \quad (9)$$

where

$$h_{ij}(q^{-1}) = K_{ij}(1+\frac{\Delta t}{T_{Iij}(1-q^{-1})})$$

$$i = 1,2; j = 1,2.$$

define

$$K_{Iij} = \frac{K_{ij}\Delta t}{T_{Iij}}$$

From equations (4) and (9)

$$\begin{aligned} u_1(t) &= u_{1c}(t-1) - K_{11}e_1(t-1) - K_{12}e_2(t-1) \\ &\quad +(K_{11}+K_{I11})e_1(t) \\ &\quad +(K_{12}+K_{I12})e_2(t) \qquad (10) \end{aligned}$$

A block diagram of the controller for a two-input single-output system is shown in Fig. 3. The output of the controller denoted by $u_1(t)$ provides the command to actuator 1 shown in Fig. 2. The implementation of the controller for actuator 2 can be obtained in a similar manner.

The results can be extended to the implementation of anti-windup strategy for a higher-order matrix controller. For example, consider a two-input two-output controller defined by equation (9) with PI and PID actions. Let

$$\begin{aligned} h_{11}(q^{-1}) &= K_{11}(1+\frac{\Delta t}{T_{I11}(1-q^{-1})} \\ &\quad +\frac{T_{D11}(1-q^{-1})}{\Delta t(1+\gamma q_{-1})}) \qquad (11) \\ h_{12}(q^{-1}) &= K_{12}(1+\frac{\Delta t}{T_{I12}(1-q^{-1})}) \qquad (12) \end{aligned}$$

The controller command $u_1(t)$ is given by

$$\begin{aligned} u_1(t) &= a_1 u_{1c}(t-1) + a_2 u_{1c}(t-2) + b_0 e_1(t) \\ &\quad +b_1 e_1(t-1) + b_2 e_1(t-2) + b_3 e_2(t) \\ &\quad +b_4 e_2(t-1) + b_5 e_2(t-2) \qquad (13) \end{aligned}$$

where $a_1, a_2, b_0, ..., b_5$ are constants that are functions of the controller parameters. The controller command $u_2(t)$ can be realized in a similar manner. The implementation provides a minimal realization of the matrix controller.

Method II

In method II the basic structure of the compensator in each path is preserved such that each path can be monitored. This is useful in on-line tuning of the matrix controller.

From equation (9), define

$$u_{11}(t) = K_{11}(1+\frac{\Delta t}{T_{I11}(1-q^{-1})})e_1(t) \quad (14)$$

$$u_{12}(t) = K_{12}(1+\frac{\Delta t}{T_{I12}(1-q^{-1})})e_2(t) \quad (15)$$

where $u_{11}(t)$ and $u_{12}(t)$ are the control commands based on errors $e_1(t)$ and $e_2(t)$ respectively. The control command to actuator 1 in the absense of limits is given by

$$u_1(t) = u_{11}(t) + u_{12}(t) \quad (16)$$

From equations (14) and (15)

$$\begin{aligned} u_{11}(t) &= u_{11}(t-1) + K_{11}e_1(t) + K_{I11}e_1(t) \\ &\quad -K_{11}e_1(t-1) \qquad (17) \\ u_{12}(t) &= u_{12}(t-1) + K_{12}e_2(t) + K_{I12}e_2(t) \\ &\quad -K_{12}e_2(t-1) \qquad (18) \end{aligned}$$

The control signal $u_1(t)$ can be written as

$$\begin{aligned} u_1(t) &= u_1(t-1) + u_{11}(t) - u_{11}(t-1) + K_{12}e_2(t) \\ &\quad +K_{I12}e_2(t) - K_{12}e_2(t-1) \qquad (19) \end{aligned}$$

Let u_{1c}, u_{11c} and u_{12c} denote control signals with saturation limits similar to equation (4). From equation (19)

$$\begin{aligned} u_1(t) &= u_{1c}(t-1) + u_{11c}(t) - u_{11c}(t-1) + K_{12}e_2(t) \\ &\quad + K_{I12}e_2(t) - K_{12}e_2(t-1) \end{aligned} \tag{20}$$

A block diagram of PI controller for a two-input single-output system is shown in Figure 4. The anti-windup strategy is based on equation (20). The path from $e_1(t)$ to $u_{11c}(t)$ has the structure of a single-input single-output PI compensator shown in Figure 1. The actuator command $u_{1c}(t)$ is computed from control signals $u_{11c}(t)$ and $u_{12}(t)$ and limiter 2. The limiter 3 is provided to assure that $u_{1c}(t) - K_{12}e_2(t) - u_{11c}(t)$ is within the limits. A block diagram for a three-input single-output controller is given in Figure 5.

In method II, the signals for the proportional and integral compensation in each path can be directly monitored and provides a way for on-line tuning of control gains.

The concept described in the paper has been demonstrated on a resaturator in the integrated gassification combined cycle (IGCC) process evaluation facility [3]

The controller was implemented on HP 1000 process computer with a disk and tape drive and HP 3497 data acquisition system.

References

1. K.J. Astrom and B. Wittenmark, *Computer Controlled Systems*, Prentice Hall, 1984.
2. R. Hanus, M. Kinnaert and J.L. Henrotte, "Conditioning Technique, a General Anti-windup and Bumpless Transfer Method", *Automatica*, pp. 729-739, 1987.
3. R.S. Baheti,"Rapid Tuning of Fuel Gas Saturator Control System using Nyquist and Bode Diagrams", *23rd IEEE Conference on Decision and Control*, Dec 10-12, 1984.

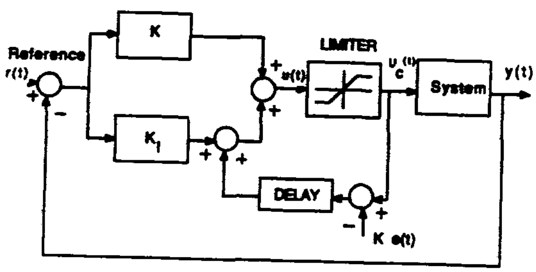

Fig. 1. Single Input Single Output Digital PI Regulator

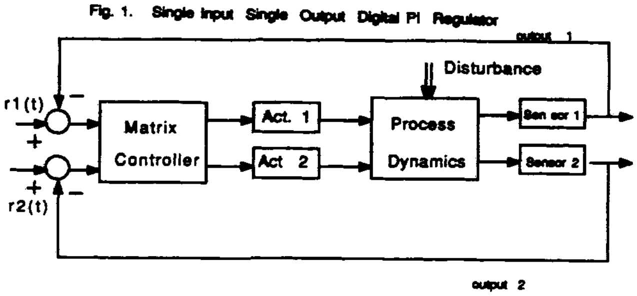

Fig. 2 A Block Diagram of Two Input Two Output Control System

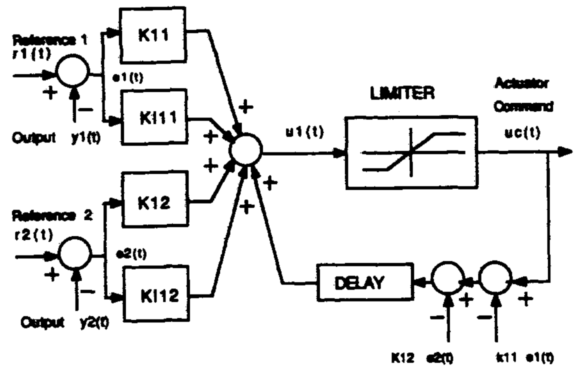

Fig. 3 Anti-windup Implementation for Two-input Two-output PI Controller (method1)

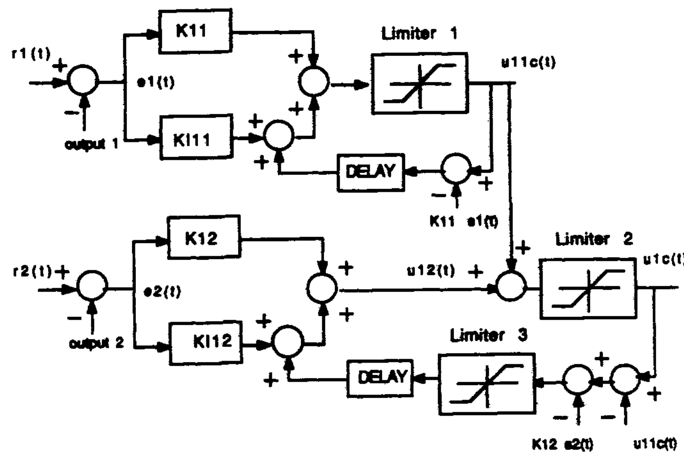

Fig. 4 Anti-windup Implementation for two-input one-output PI Controller (Method II)

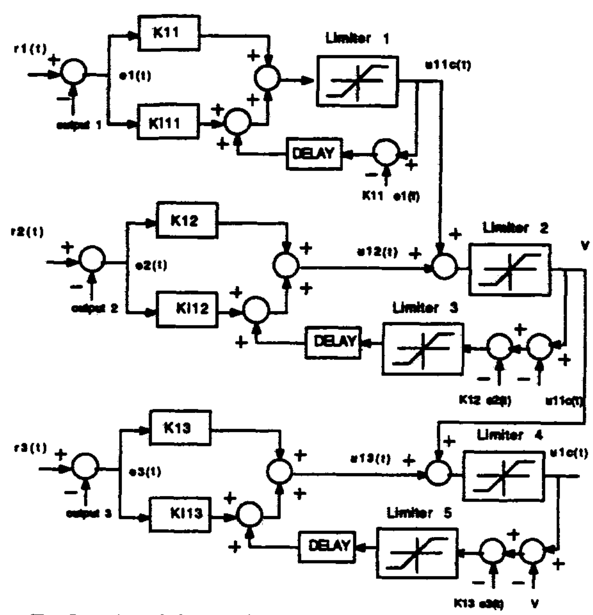

Fig. 5 Anti-windup Implementation for three-input one-output PI Controller (Method II)