ISSN: 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

Constrained Cascade control For Parallel Cascade system

M.Bharathi*,C.Selvakumar**

*HOD, Department of Electronics and Instrumentation, Bharath University, Chennai-600073, India

**Prof & Head, St.Joseph's College Of Engineering, Chennai-119, India

ABSTRACT: Output constraints usually occur when the number of manipulated variables is smaller than the number of output variables. In this case, every output cannot have its own set point. Some of the outputs can then only be constrained to a specific range of values instead of being associated to a specific set point value. The respect of the constraints for these outputs has priority over the other outputs with set points. One of the objective of this section is to develop methods applicable to a one-input and multiple-output processes where one output must reach a given set point as long as the other outputs lie between their lower and upper limits.

KEY WORDS: manipulated variables, set point,Cascade system

I.INTRODUCTION

A possible solution is override control (Shinskey, 1984). This method is a selective control strategy where the manipulated variable is selected as the largest or the smallest output value of the PID controllers. One controller is tuned to make the process output reach its set point while the other controllers are each set to bring the process output to a constraint limit without overshooting. To prevent reset windup, the integration part of the controllers is achieved by a positive feedback of the effective manipulated variable. Some authors (Giles and Gaines, 1978) have suggested that, instead of making both controller outputs track the effective manipulated variable, better results are achieved by making both controller integrators tracking each other.

This section studies properties of cascaded control loop to achieve constrained control. Cascade controllers (Boyce and Brosilow, 1996) are described and their qualities and drawbacks are highlighted. A variation of this controller, called pseudo-cascade (Lestage *et al.*,1997), is presented. A simulated example compares both methods.

II.SERIAL CASCADE CONTROLLER

Serial cascade controllers are well known and widely used. A serial cascade controller is shown in figure 1. The final control variable is $y2$ but an intermediate measurement $y1$ is available. Controller $G_{C1}(s)$ regulates output $y1$. The controller $G_{C2}(s)$ manipulates the first controller set point $r1$ to regulate the final control variable $y2$. The main quality of this controller structure is the ability to cancel disturbance $p1$ faster than a single loop controller (Caldwell, 1959). The saturation block is a device used to constrain the amplitude of the intermediate output variable $y1$ through its set point $r1$. Due to possible controller output saturation, both controllers must feature anti-reset wind-up protection.

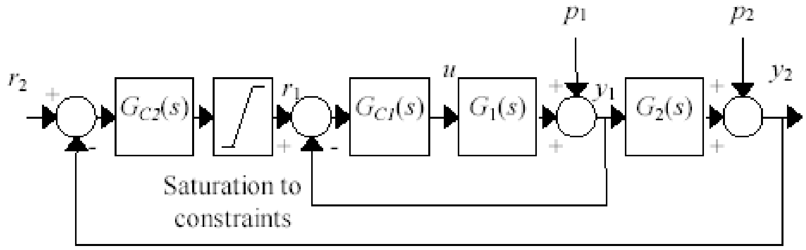

Fig.1 Serial cascade controller

III.PARALLEL CASCADE CONSTRAINT HANDLING METHOD

Serial process representation is not always possible due to the physical nature of the process. Figure 2 shows the same controller as the one of fig.1 with a different process represented by parallel transfer functions. The goal of the system is to make *y*2 reach the set point *r*2 as long as the constraints on *y*1 are respected. The inner-loop controller, $G_{C1}(s)$, is used to regulate the constrained output, *y*1. $G_{C1}(s)$ is tuned to avoid overshooting of the constrained variable *y*1. The saturation insures that the inner loop set point respects the constraint imposed to *y*1. The second controller, $G_{C2}(s)$, is tuned to regulate the output *y*2 to its set point. Because of the saturation, controllers must again have the ability to prevent integral wind-up. When more than one output is constrained, additional cascade loops are nested. Constraints are then respected with the priority given from the inner loop toward the outer loops.

In open-loop, the disturbance *p*1 does not affect the output *y*2. However, in closed loop, the output *y*2 becomes sensitive to the disturbance *p*1. The transfer function from *p*1 to *y*2, calculated is

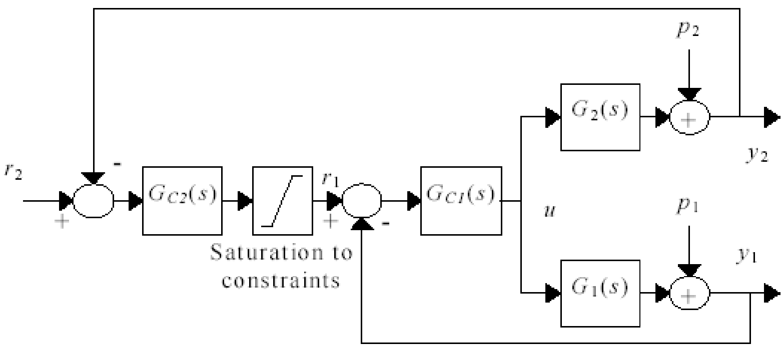

Fig.2 Parallel cascade method

$$\frac{y_2(s)}{p_1(s)} = \frac{-G_{C1}(s)G_{C2}(s)}{1+G_{C1}(s)G_1(s)+G_{C1}(s)G_{C2}(s)G_2(s)} \quad (1)$$

The closed-loop transfer function from *r*2 to *y*2 is then:

$$\frac{y_2(s)}{r_2(s)} = \frac{G_{C1}(s)G_{C2}(s)G_2(s)}{1+G_{C1}(s)G_1(s)+G_{C1}(s)G_{C2}(s)G_2(s)} \quad (2)$$

The previous equations state that the tuning of *G*C2(*s*), based on *y*2(*s*)/*r*2(*s*) is function of *GC*1(*s*), *G*1(*s*) and *G*2(*s*). Thus any modification of the inner control loop requires a new tuning for *GC*2(*s*). The outer control loop is sensitive to uncertainties on both *G*1(*s*) and *G*2(*s*). If more than two loops are cascaded, complexity raises since the outer controllers are dependent of the transfer functions of every other inner controllers and their associated process transfer functions.

IV.PSEUDO-CASCADE CONSTRAINT HANDLING METHOD

An innovative and simple cascade strategy is depicted in figure 3. Again, a first controller *GC*1(*s*) is used to control the constrained output. The set point of this controller is limited to the permissible value of the output *y*1. In this structure, it can be seen that when the saturation is not active, the positive feedback *y*1 cancels the negative one. The method is called pseudocascade since the inner feedback loop is not effective unless saturation occurs. The transfer function of the second controller is $G_{C2}(s)\ G_{C1}^{-1}(s)$. Since $G_{C1}^{-1}(s)$ cancels $G_{C1}(s)$, then GC2(s) is tuned with respect to the process $G_2(s)$. The resulting closed-loop transfer function is :

$$\frac{y_2(s)}{r_2(s)} = \frac{G_{C2}(s)G_2(s)}{1+G_{C2}(s)G_2(s)} \quad (3)$$

The tuning of $G_{C2}(s)$ is independent of the inner loop controller $G_{C1}(s)$ and inner loop process $G_1(s)$. This property allows the nesting of a great number of control loops with simple and independent tuning of each loop. This provides easier maintenance since the tuning of an inner loop does not require any correction to the other loops. Another advantage of this structure over the parallel cascade is that, when the saturation is not active, disturbances $p1$ and noise occurring on $y1$ are cancelled and are not fed back into the inner loop. This property makes the pseudo-cascade strategy less prone to react to disturbances on $y1$ when the constraint is not active. This property will be later illustrated by an example.

Since the system contains saturating elements, care must be taking to prevent nonlinear phenomena such as integrator windup, limit cycles operation or signal saturation due to noise. The integrator windup problem is easily solved when both $GC2(s)$ and $GC1(s)$ have an integrator. No anti-reset windup feature is required in the $GC2(s)GC1$ -1(s) controller because the integrator of the controller $GC2(s)$ is cancelled by $GC1$ -1(s). It is however important to implement the minimum (simplified) realization of $GC2(s)GC1$ -1(s) into the process computer. The presence of noise in the system can result in unexpected behavior due to signal clipping by the saturation element. The noise makes the system continuously switch between the inner and outer loop control. The stationary response is then different from the noiseless steady-state behavior.

V.SENSITIVITY TO DISTURBANCES

In order to compare the responses of the parallel cascade and the pseudo-cascade methods to a disturbance $p1$, a simulated process with the following parallel transfer functions is used:

$$G_1(s) = \frac{1}{(1+4s)^2} \tag{4}$$

$$G_2(s) = \frac{1}{(1+12s)(1+5s)} \tag{5}$$

For both methods, controllers are tuned to get the following closed loop transfer functions:

$$H_1(s) = \frac{1}{(1+4s)^2} \tag{6}$$

$$H_2(s) = \frac{1}{(1+7s)^2} \tag{7}$$

Pole-zero cancellation method led to the following controllers for the parallel cascade method:

$$G_{C1}(s) = \frac{(1+4s)^2}{8s(1+2s)} \tag{8}$$

$$G_{C2}(s) = \frac{(1+12s)(1+5s)}{7s(2+7s)} \tag{9}$$

Similarly, pseudo-cascade controllers are:

$$G_{C1}(s) = \frac{(1+4s)^2}{8s(1+2s)} \tag{10}$$

$$G_{C2}(s) = \frac{(1+12s)(1+5s)}{7s(2+7s)} \tag{11}$$

Simulations shown in figure 3 illustrate a set point change from $r2$=0 to $r2$=1 at time 0. This set point is achieved without error or constraint problem. At time 100 sec, a disturbance $p1$=0.25 step change occurs. The parallel cascade configuration takes action on this disturbance even if there is no constraint violation. Since there is no constraint violation, pseudo-cascade saturation is not active and the pseudo-cascade algorithm takes no unnecessary action. In this case, the

ISSN: 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

pseudocascade method is less sensitive to disturbance. At time 150 sec., the set point value is changed to 2. The set point is not reached due to the constraint on $y1$. Both methods achieve the same set point and constraints dynamics when same closed-loop specifications are specified. The pseudo-cascade is however less sensitive to the disturbance $p1$ when the constraint is not active.

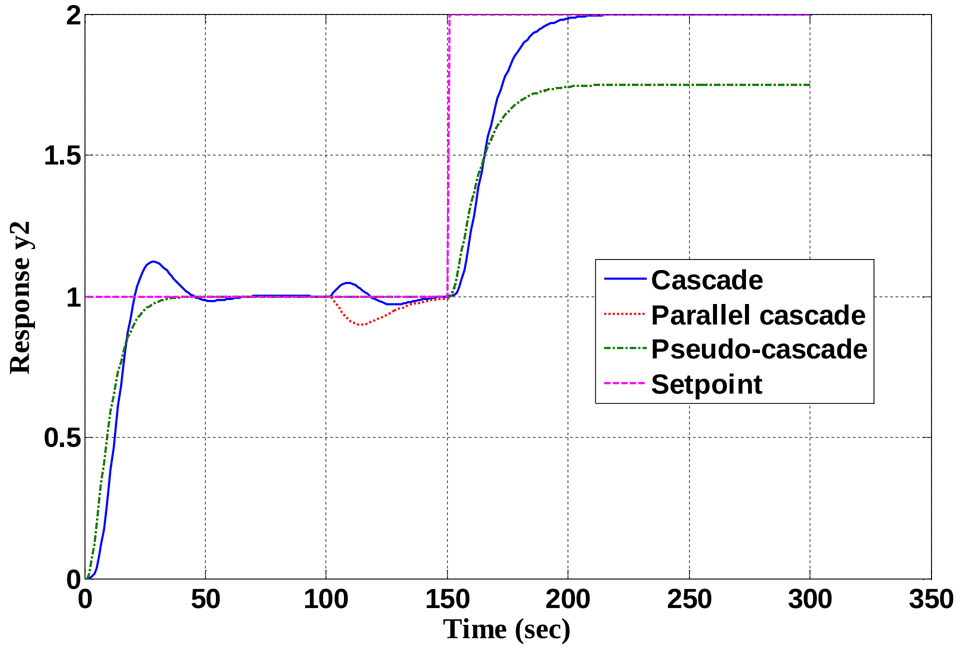

Fig.3 comparison of cascade, parallel cascade, pseudo cascade for set point

VI.CONCLUSION

The problem of maintaining an output to a set point while keeping another output constrained within a given limit has been assessed. A first method based on cascade control has shown to be sensitive to disturbances on the constrained variable even when there is no limit transgression. A modification to the cascade method in order to correct this problem has led to the pseudo-cascade method. The pseudo-cascade method allows the nesting of a large number of control loops with simple and independent tuning for each loop.

Pseudo-cascade can also be applied to multivariable processes with perfect inverted decoupling. Decouplers and feedback controllers can be cancelled when no constraints are active in order to reconstruct the manipulated variables. These reconstructed manipulated variables can then be used to achieve other control objectives.

REFERENCES

1. William L. Luyben, (1987), Derivation of Transfer Functions for Highly Nonlinear Distillation Columns, *Ind. Eng. Chem. Res.* , Vol.26, 2490-2495
2. Yu, Cheng-Ching (1999), *Autotuning of PID Controllers: Relay Feedback Approach, Springer verlag*, Berlin. KG.
3. Yu, C.-C., (1988) “Design of Parallel Cascade Control for Disturbance-Rejection”, *AIChE J.*, 34, pp.833-1838.
4. Ziegler, J.G. and Nichols, N.B. (1942), “Optimum settings for automatic controllers”, *Transactions of the ASME*, November, pp. 759-768.