Abstract: This paper addresses first the problem of constraints handling of a system with one input and multiple outputs where one output must reach a given set point and the other outputs must lie between lower and an upper limits. Three algebraic solutions based on cascade control are outlined. One method is the traditional cascade controllers applied to serial transfer functions. The second is cascade controllers applied to parallel transfer function processes. The latter method shows sensitivity to disturbance and tuning of inner loops. A third innovative method, called pseudo-cascade controller, is introduced for parallel transfer functions. The new method allows independent tuning of the controllers and requires no special anti-reset windup feature. An extension is also given for decoupled two input processes. A simulated example and a distributed control design for an industrial application are given to illustrate the proposed methods.

Keywords: Cascade control, Constraints, Decoupling, Saturation.
1. INTRODUCTION

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 paper 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.

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 paper 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. An application is given for a two input, four output grinding circuit process.
2. 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 $y_2$ but an intermediate measurement $y_1$ is available. Controller $G_{C1}(s)$ regulates output $y_1$. The controller $G_{C2}(s)$ manipulates the first controller set point $r_1$ to regulate the final control variable $y_2$. The main quality of this controller structure is the ability to cancel disturbance $p_1$ faster than a single loop controller (Caldwell, 1959). The saturation block is a device used to constrain the amplitude of the intermediate output variable $y_1$ through its set point $r_1$. Due to possible controller output saturation, both controllers must feature anti-reset wind-up protection.

3. 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 without saturation, is:

$$\frac{y_2(s)}{p_1(s)} = \frac{-G_{C_1}(s)G_{C_2}(s)}{1 + G_{C_1}(s)G_1(s) + G_{C_1}(s)G_{C_2}(s)G_2(s)}$$  \hspace{1cm} (1)$$

The closed-loop transfer function from $r_2$ to $y_2$ is then:

$$\frac{y_2(s)}{r_2(s)} = \frac{G_{C_1}(s)G_{C_2}(s)G_2(s)}{1 + G_{C_1}(s)G_1(s) + G_{C_1}(s)G_{C_2}(s)G_2(s)}$$  \hspace{1cm} (2)$$

The previous equations state that the tuning of $G_{C_2}(s)$, based on $y_2(s)/r_2(s)$ is function of $G_{C_1}(s)$, $G_1(s)$ and $G_2(s)$. Thus any modification of the inner control loop requires a new tuning for $G_{C_2}(s)$. 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.

4. PSEUDO-CASCADE CONSTRAINT HANDLING METHOD

An innovative and simple cascade strategy is depicted in figure 3. Again, a first controller $G_{C_1}(s)$ 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 pseudo-cascade since the inner feedback loop is not effective unless saturation occurs. The transfer function of the second controller is $G_{C_2}(s)G_{C_1}^{-1}(s)$. Since $G_{C_1}^{-1}(s)$ cancels $G_{C_1}(s)$, then $G_{C_2}(s)$ is tuned with respect to the process $G_2(s)$. The resulting closed-loop transfer function is:
\[
\frac{y_2(s)}{r_2(s)} = \frac{G_{C2}(s)G_z(s)}{1 + G_{C2}(s)G_z(s)}
\]  

(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 \( p_1 \) and noise occurring on \( y_1 \) 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 \( y_1 \) 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 \( G_{C2}(s) \) and \( G_{C1}(s) \) have an integrator. No anti-reset windup feature is required in the \( G_{C2}(s)G_{C1}^{-1}(s) \) controller because the integrator of the controller \( G_{C2}(s) \) is cancelled by \( G_{C1}^{-1}(s) \). It is however important to implement the minimum (simplified) realization of \( G_{C2}(s)G_{C1}^{-1}(s) \) 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.

5. EXAMPLE: SENSITIVITY TO DISTURBANCES
In order to compare the responses of the parallel cascade and the pseudo-cascade methods to a disturbance $p_1$, a simulated process with the following parallel transfer functions is used:

\[
G_1(s) = \frac{1}{(1+2s)^2} \quad (4)
\]

\[
G_2(s) = \frac{1}{(1+10s)(1+3s)} \quad (5)
\]

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

\[
H_1(s) = \frac{1}{(1+2s)^2} \quad (6)
\]

\[
H_2(s) = \frac{1}{(1+5s)^2} \quad (7)
\]

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

\[
G_{C1}(s) = \frac{(1+2s)^2}{4s(1+s)} \quad (8)
\]

\[
G_{C2}(s) = \frac{(1+10s)(1+3s)}{10s(1+2.5s)} \quad (9)
\]

Similarly, pseudo-cascade controllers are:

\[
G_{C1}(s) = \frac{(1+2s)^2}{4s(1+s)} \quad (10)
\]

\[
G_{C2}(s) = \frac{(1+10s)(1+3s)}{10s(1+2.5s)} \quad (11)
\]
Simulations shown in figure 4 illustrate a set point change from $r_2=0$ to $r_2=50$ at time 0. This set point is achieved without error or constraint problem. At time 40 sec, a disturbance $p_1=25$ 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 pseudo-cascade method is less sensitive to disturbance. At time 80 sec., the set point value is changed to 100. The set point is not reached due to the constraint on $y_1$.

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 $p_1$ when the constraint is not active.

6. EXTENSION TO MULTIVARIABLE PROCESSES WITH DECOUPLER

Figure 5 shows a multivariable process with a perfect decoupler and single-loop controllers.

The process transfer function matrix is:

$$
\begin{bmatrix}
y_1(s) \\
y_2(s)
\end{bmatrix} =
\begin{bmatrix}
G_{11}(s) & G_{12}(s) \\
G_{21}(s) & G_{22}(s)
\end{bmatrix}
\begin{bmatrix}
u_1(s) \\
u_2(s)
\end{bmatrix}
$$

(12)

The perfect decoupler is built in the inverse form where each branch of the decoupler is fed before the other branch pickup point (Gagnon et al., 1998). Its transfer function is:

$$
\begin{bmatrix}
u_1(s) \\
u_2(s)
\end{bmatrix} =
\begin{bmatrix}
G_{11}(s)G_{22}(s) - G_{12}(s)G_{22}(s) \\
-G_{11}(s)G_{22}(s) + G_{12}(s)G_{21}(s)
\end{bmatrix}
\begin{bmatrix}
u'_1(s) \\
u'_2(s)
\end{bmatrix}
$$

(13)

This decoupler is realizable only if the decoupler transfer function matrix is open-loop stable. It can be shown that the transfer function matrix of the decoupler in series with the process is:
For the perfect decoupler in the inverse form, single-loop controllers are tuned on the single direct transfer functions $G_{11}(s)$ and $G_{22}(s)$. Controllers are then properly tuned even if one control loop is open.

As an extension of fig. 3, if the set points $r_1$ and $r_2$ are constrained variables, a pseudo-cascade controller structure can be used to modify the control strategy in order to cancel the $y_1$ and $y_2$ feedback and to cancel the decoupler when the constraints are not active. Figure 6 shows that the decoupling can be achieved at the controller input with a suitable modification to the decoupler transfer functions. Inversion of the controllers at the inputs permits to reconstruct the manipulated variables $u_1$ and $u_2$ which can then be used by two other controllers to achieve control objectives with less priority than the constraints while respecting the allowable $r_1$ and $r_2$ set point values.

When saturation is not active, the variables $\hat{u}_i$ differ from $u_i$ only by additive constants equal to the value of the controller integrators. Therefore, the variable $\hat{u}_i$ can be used as manipulated variables by another controller to regulate additional variables not shown in the figure.

Usually, the controllers have an integrator. The inverses of the controllers then have a derivative function. To prevent integrator problems, the inverse of $G_{Ci}$ must be embedded with the controller using $\hat{u}_i$ as a manipulated variable in order to build a minimal representation of the system by canceling the integrator and derivative term of the serial composition of the two transfer functions.

\[
\begin{bmatrix}
y_1(s) \\
y_2(s)
\end{bmatrix} = \begin{bmatrix}
G_{11}(s) & 0 \\
0 & G_{22}(s)
\end{bmatrix} \begin{bmatrix}
u'_1(s) \\
u'_2(s)
\end{bmatrix}
\]  

(14)
The above strategy leads to a distributed controller with constraints handling. Perfect decouplers are however not always realizable.

7. DESIGN OF A MULTI-LOOP PSEUDO-CASCADE CONTROLLER FOR A GRINDING CIRCUIT

The grinding circuit is used in mineral processing to reduce the size of ore particles in order to liberate valuable mineral from not valuable one. A typical rod mill, ball mill circuit with hydrocyclones classification is depicted in fig. 7. For control purposes, the grinding circuit can be considered as a two-input four-output process. The inputs are:

- $u_{PBW}$ Pump box water flow rate (m$^3$/h)
- $u_{RMF}$ Fresh ore rod mill feed rate (t/h)

The four outputs are:

- $y_{M45}$ Fraction of particle size smaller than 45 um (%)
- $y_{COFD}$ Hydrocyclone overflow pulp density (%)
- $y_{PBL}$ Pump box level (%)
- $y_{CL}$ Circuit circulating load (t/h)

In order of decreasing importance, process operating goals are:

1) Achieve the $r_{M45}$ set point on $y_{M45}$.
2) Keep $y_{COFD}$ between 48 and 52%.
3) Keep $y_{PBL}$ between 15 and 85%.
4) Achieve the $r_{CL}$ set point on $y_{CL}$.
Table 1 shows the process transfer functions. Figure 8 shows the first proposed control strategy. The first objective is achieved through the control of $y_{M45}$ with the $u_{PBW}$ manipulated variable. This choice comes from its fast response due to an important negative zero in the transfer function. The remaining manipulated variable $u_{RMF}$ is used to maintain remaining control objectives in order of importance.

The set points $r_{COFD}$ and $r_{M45}$ are achieved with perfect decoupling due to $D_{M45}(s)$ and $D_{COFD}(s)$. A saturation element ensures that the COFD set point is within its constraints. When there is no saturation, the measured $y_{COFD}$ and the output of $D_{COFD}(s)$ do not affect the input of $G_{C1}(s)$. The $r_{PBL}$ set point is then active with the $D_{PBL}(s)$ decoupler. Another saturation element ensures correct PBL set point. When this saturation is not effective, a second pseudo-cascade loop allows CL to be controlled. With perfect decouplers, controllers are tuned independently from each other on the single transfer function from the manipulated variable to the controlled output.

Because of the presence of right-half plane zeros, perfect decoupling is not realizable for this specific case. A new control strategy with a one direction decoupler $D_{M45}(s)$ is used to cancel the effect of $u_{RMF}$ on $y_{M45}$ as proposed in fig. 9. PI controllers and the decoupler transfer functions are shown in table 2. They are tuned by using a dominant pole-zero cancellation method on the resulting transfer function formed by the parallel effect of $u_{RMF}$ and the decoupler. The gains of the controllers have been adjusted to avoid overshoots at the process output and to obtain manipulated variables with acceptable amplitudes. Resulting controllers $G_{C2}(s)G_{C1}^{-1}(s)$ and $G_{C3}(s)G_{C2}^{-1}(s)$ are simple gain and thus require no anti-reset windup protection to account for the saturation effect.
Simulation results are shown in figure 10. The simulation sequence shows a $y_{CL}$ set point change at time 0. The set point is reached without problem or constraint violation. At time 5h, an additive output disturbance occurs on $y_{CL}$. The control strategy maintains the $r_{M45}$ set point and $y_{CL}$ cannot recover from this disturbance due to the upper constraint on $y_{PBL}$. When a second disturbance occurs on $y_{PBL}$ at time 10h, the active constraint is now the lower $y_{COFD}$ limit. At time 15h, a set point change is made on $y_{M45}$. The set point change response is good but the coupling causes a long acting disturbance on the other controlled variables.

8. 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


Gagnon, E., Pomerleau A. and Desbiens, A., (1998), Simplified, Ideal or Inverted Decoupling?, Accepted for publication in *ISA Transactions*. 
Fig. 1: Serial cascade controller

Fig. 2: Parallel cascade method.

Fig. 3: Pseudo-cascade method.
Fig. 4. Simulation.

Fig. 5: Multivariable process
Fig. 6: Pseudo-cascade multivariable process.

Fig. 7. A rod mill, ball mill grinding circuit with hydrocyclone classification

Fig. 8: First proposed grinding circuit control strategy.
Fig. 9: Final grinding circuit control strategy.
Fig. 10: Simulation results.
Table 1: Process transfer functions (time constants in seconds)

<table>
<thead>
<tr>
<th></th>
<th>( u_{RMF} )</th>
<th>( u_{PBW} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_{COFD} )</td>
<td>( \frac{0.0255(1 - 5600s)e^{600s}}{(1 + 5300s)(1 + 750s)} )</td>
<td>( \frac{-0.14(1 + 4050s)}{(1 + 3200s)(1 + 60s)} )</td>
</tr>
<tr>
<td>( Y_{M45} )</td>
<td>( \frac{-0.2(1 - 900s)}{(1 + 5200s)(1 + 750s)} )</td>
<td>( \frac{0.012(1 + 39500s)}{(1 + 44000s)(1 + 50s)} )</td>
</tr>
<tr>
<td>( Y_{CL} )</td>
<td>( \frac{13.8}{(1 + 5700s)(1 + 400s)} )</td>
<td>( \frac{4.2(1 - 700s)}{(1 + 5000s)(1 + 5s)} )</td>
</tr>
<tr>
<td>( Y_{PBL} )</td>
<td>( \frac{5.749}{(1 + 5500s)(1 + 210s)} )</td>
<td>( \frac{1.962}{1 + 4700s} )</td>
</tr>
</tbody>
</table>
## Table 2: Controllers transfer functions

<table>
<thead>
<tr>
<th>Transfer Function</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_{CM_{45}}(s) )</td>
<td>( 160 \frac{(1 + 4400s)}{4400s(1 + 39500s)} )</td>
</tr>
<tr>
<td>( G_{C1}(s) )</td>
<td>( -6 \frac{(1 + 39500s)}{39500s} )</td>
</tr>
<tr>
<td>( G_{C2}(s) )</td>
<td>( 0.3 \frac{(1 + 39500s)}{39500s} )</td>
</tr>
<tr>
<td>( G_{C3}(s) )</td>
<td>( 0.17 \frac{(1 + 39500s)}{39500s} )</td>
</tr>
<tr>
<td>( D_{M_{45}}(s) )</td>
<td>( 0.2 \frac{(1 + 4400s)(1 - 900s)}{0.012(1 + 39500s)(1 + 5200s)} )</td>
</tr>
<tr>
<td>( G_{C2}(s) G_{C1}^{-1}(s) )</td>
<td>( -\frac{0.3}{6} )</td>
</tr>
<tr>
<td>( G_{C3}(s) G_{C2}^{-1}(s) )</td>
<td>( \frac{0.017}{0.3} )</td>
</tr>
</tbody>
</table>