

Development of a Jacobian module for advanced pulp and paper simulation and control

MOHAMED CHBEL AND LUC LAPERRIÈRE

ABSTRACT: Pulp and paper processes frequently present nonlinear behavior, which means that process dynamics change with the operating points. These nonlinearities can challenge process control. PID controllers are the most popular controllers because they are simple and robust. However, a fixed set of PID tuning parameters is generally not sufficient to optimize control of the process. Problems related to nonlinearities such as sluggish or oscillatory response can arise in different operating regions. Gain scheduling is a potential solution. In processes with multiple control objectives, the control strategy must further evaluate loop interactions to decide on the pairing of manipulated and controlled variables that minimize the effect of such interactions and hence, optimize controller's performance and stability. Using the CADSIM Plus™ commercial simulation software, we developed a Jacobian simulation module that enables automatic bumps on the manipulated variables to calculate process gains at different operating points. These gains can be used in controller tuning. The module also enables the control system designer to evaluate loop interactions in a multivariable control system by calculating the Relative Gain Array (RGA) matrix, of which the Jacobian is an essential part.

Application: PID controller performance can be optimized using a Jacobian module developed inside a commercial pulp and paper simulator.

Pulp and paper processes (and chemical processes in general) exhibit nonlinear behavior; that is, process gains and time constants change with the operating points. These variations have a great impact on PID controller performance and stability and can cause problems in both product quality and operating costs.

The proper tuning of PID parameters is important to maintain acceptable performance over the operating interval to ensure stability, especially for nonlinear processes. The choice of robust PID parameters will not guarantee optimal controller performance.

Gain scheduling can be a solution for nonlinear processes where a set of different PID parameters are used to ensure optimal performance over the entire operating region. The tuning parameters of the PID controller can be calculated by different tuning methods [1–3]. In pulp and paper, lambda tuning is the most popular. The objective is to achieve a smooth response to set point changes. PID tuning parameters are calculated from the process model by performing a bump test on controller output and then observing the process response. Process gain, time constant, and dead time can then be estimated and a simple dynamic process model can be obtained.

Generally, the process gain is the most important tuning parameter [4] and estimation of the process gain at different operating points can become very valuable. Gain scheduling is a strategy that makes use of different gain values at different operating points [5, 6].

In multiple inputs multiple outputs (MIMO) systems, mul-

multiple variables must be controlled by manipulating several variables. Control objectives can each be considered separately as long as they don't interact. The pairing of controlled and manipulated variables that minimize loop interactions can have a big influence on the overall stability and performance of the control system. Bristol [7] has published a quantitative method that calculates the relative gain array (RGA) from the steady state gain matrix. The proper analysis of each RGA matrix entry is a valuable tool for choosing the best pairing that minimizes interactions.

In light of this need to tune parameters, a design tool that could evaluate process gains in simulation software would be useful. The Jacobian module we describe can evaluate gains between sets of controlled variable y_i and manipulated variables x_j to be used in controller tuning. It also serves as the basis for the computation of the RGA for MIMO systems design.

JACOBIAN MODULE

CADSIM Plus™ is commercial simulation software in which custom user modules, called dynamically linked modules (DLM), can be developed. Using a pre-defined syntax, a C++ programmer can incorporate his own computations that are encapsulated in his own DLM. After compilation and linking, the DLM becomes part of the software and can interact with any other simulation module. We developed a Jacobian module as a DLM in CADSIM Plus™.

The Jacobian DLM requires the user to specify a number of variables in the flow sheet drawing interface, which repre-

sents the whole process to be simulated. The Jacobian module will then compute some other variables during simulation.

The independent variables (manipulated x_i) and dependant variables (controlled y_i) are the most important. The user must navigate the flow sheet drawing interface and specify which process variables are manipulated and which are controlled. For example, if the flow sheet drawing interface represents a tank with two inlets, an overflow, and an outlet and the Jacobian module is to be used to design a tank level controller, then the user would probably specify the outlet flow as the independent (manipulated) variable and the tank level as the dependant (controlled) variable. In the flow sheet drawing interface, these variables of the tank module icon would be connected to the Jacobian module icon.

Other important variables to be specified in configuring the Jacobian module are:

- Settling time, which is the time required for the controlled variable y_i to reach its final value in steady state after a bump on the manipulated variable x_i has been performed; and
- Bump factor, which represents the amplitude of excitation (bump) of the manipulated variable x_j , expressed in Eq. 1:

$$x_i(Bumped) = x_i(initial\ value) * (1 + Bump\ factor) \quad (1)$$

For integrating systems like a tank, the settling time should be specified carefully to avoid such situations as overflow or dry out of the tank.

Jacobian module computations

After all necessary variables have been specified, the Jacobian module can be run during simulation and compute useful information, in the form of calculated (as opposed to the earlier specified) variables.

As the module name suggests, a first set of computed variables are the entries of the actual Jacobian matrix itself, representing the partials between each dependant and independent variable pair (Eq. 2):

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1m} \\ J_{21} & J_{22} & \cdots & J_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ J_{n1} & J_{n2} & \cdots & J_{nm} \end{bmatrix} \quad (2)$$

where:

J = Jacobian matrix

x_i = independent variable, $i = 1$ to m

y_i = dependant variable, $i = 1$ to n

The Jacobian matrix is obtained during simulation after a series of bumps have been performed on the set of specified independent variables $\mathbf{x}_i$. Each Jacobian entry is simply com-

puted according to Eq. 3:

$$\frac{\partial y_i}{\partial x_i} \cong \frac{\Delta y_i}{\Delta x_i} = \frac{y_{i(final)} - y_{i(initial)}}{x_{i(initial)} * bump_factor} \quad (3)$$

Once the Jacobian matrix has been obtained, the module also performs a series of matrix computations to obtain the RGA matrix (Eq. 4):

$$RGA = J \otimes (J^{-1})^T = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2n} \\ \dots & \dots & \dots & \dots \\ \lambda_{n1} & \lambda_{n2} & \lambda_{n3} & \lambda_{nn} \end{bmatrix} \quad (4)$$

where:

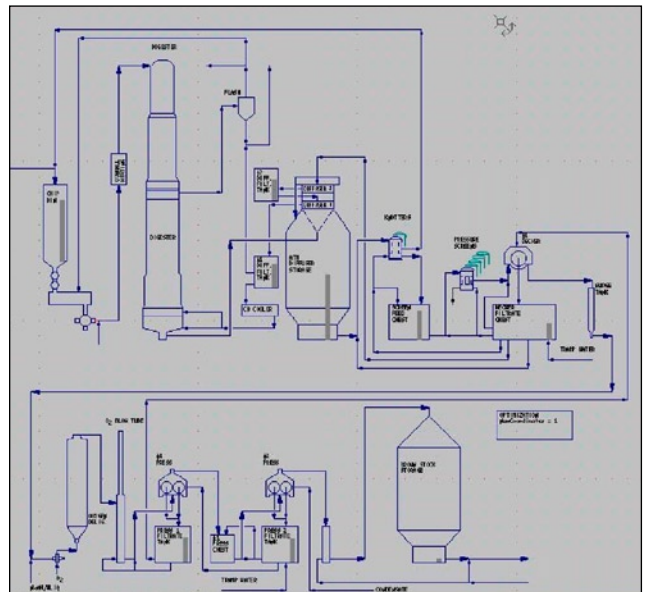
λ_{ij} = Relative gain between dependant variable y_i and manipulated variable x_j in a MIMO system.

In summary, after a number of specified variables have been provided to configure the Jacobian DLM in the drawing interface, a number of calculated variables are obtained as a result of the execution of this module during simulation. These calculated variables become valuable information for the control system designer, assuming simulation is used during the design phase.

GAIN SCHEDULING APPLICATION

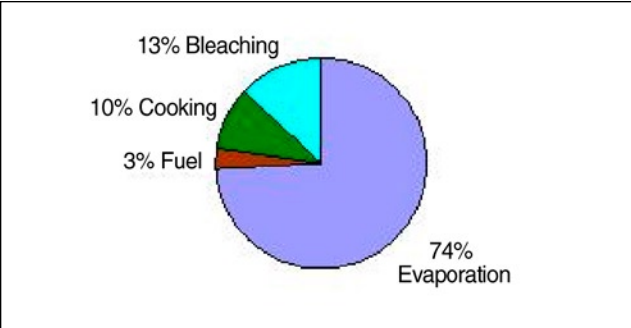
Figure 1 shows a simulation of a brown stock washing system [8]. In this system, the washing water flow and dilution factor are important parameters for achieving minimal washing costs. Although high dilution factors minimize the cost related to the use of chemicals and bleaching, they also increase evaporating costs, which are predominant (**Fig. 2**).

The process dynamics are such that different optimal dilution factors exist that minimize washing costs for different op-

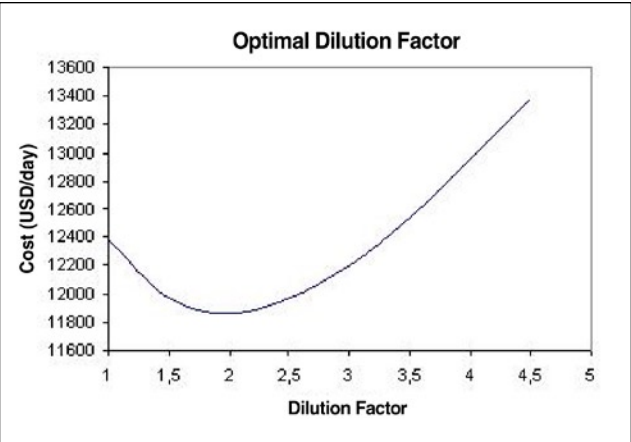


1. Simulation flow sheet of the brown stock washing system.

PROCESS CONTROL



2. Partition of washing costs components for a dilution factor of 2.



3. Optimal dilution factor (1.96) for a washed pulp production rate of 2000 t/day at 50% infeed chips consistency.

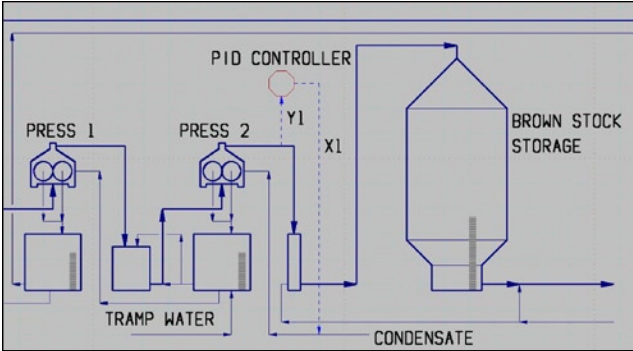
erating conditions (**Fig. 3**). Our first task was then to find optimal dilution factor for different chip feed consistencies from 45%–55% (**Table I**). The resulting perturbations in production rates have a great impact on washing control performance.

These optimal values were obtained by running the optimization module of the CADSIM Plus™ software [9].

Modifications to the simulation

The original simulation in Fig. 1 has been modified to implement a PI controller at the final washing stage (**Fig. 4**). For demonstration purposes it is assumed that this controller measures the dissolved solids in the washed pulp (y_1 in the figure) and manipulates the fresh washing water accordingly (x_1 in the figure). The set point on dissolved solids content is chosen so that the optimal dilution factor for each consistency is achieved as per Table I. We observed that an approximately constant rate of 16.4 kg of dissolved solids per ton of dry pulp allows an optimal wash for all operating points in Table I.

The Jacobian module can then be used to compute the controller gain with the lambda tuning method. The settling time for this simulation is also very important, because it takes about 4000 min of simulation time (about 1 min in real time) to stabilize the quantity of dissolved solids when a perturbation in the fresh water flow is input. The choice of 5000 min as settling time for the Jacobian module was largely sufficient



4. Adding a dissolved solids controller to the original simulation.

Consistency (%)	Dry flow softwood (t/j)	Optimal dilution factor (DF)
55	2444	2.30
54	2348	2.24
53	2255	2.22
52	2167	2.10
51	2082	2.03
50	2000	1.95
49	1922	1.88
48	1846	1.80
47	1774	1.72
46	1704	1.63
45	1636	1.53

I. Optimal dilution factor for different chips consistencies and production rates.

to compute the gain.

Figure 5 shows that a 10% decrease on fresh water at the last washing stage (from 52.16 to 46.94 l/s) increases the dissolved solids content of the washed pulp by 28% (from 16.4 to 20.92 kg/t of dry pulp). Therefore, the gain calculated by the Jacobian module is about -0.867 (kg/t/l/s) and the time constant is about 300 min.

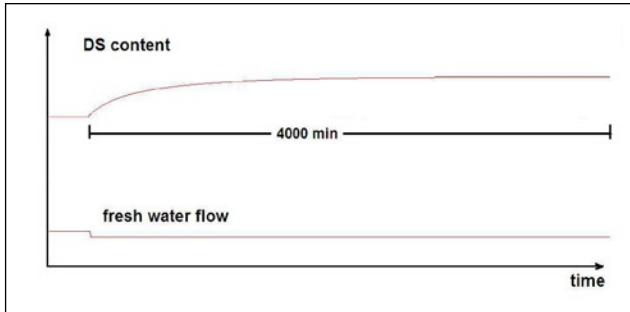
The lambda factor is chosen so that the response of the controller is fast and stable. Using the lambda tuning equation for a first order behavior, the PI tuning parameters are calculated (Eq. 5a):

$$K_c = \frac{\tau}{K_p \lambda} = \frac{300}{-0.867 * 60} = -5.76 \quad (5a)$$

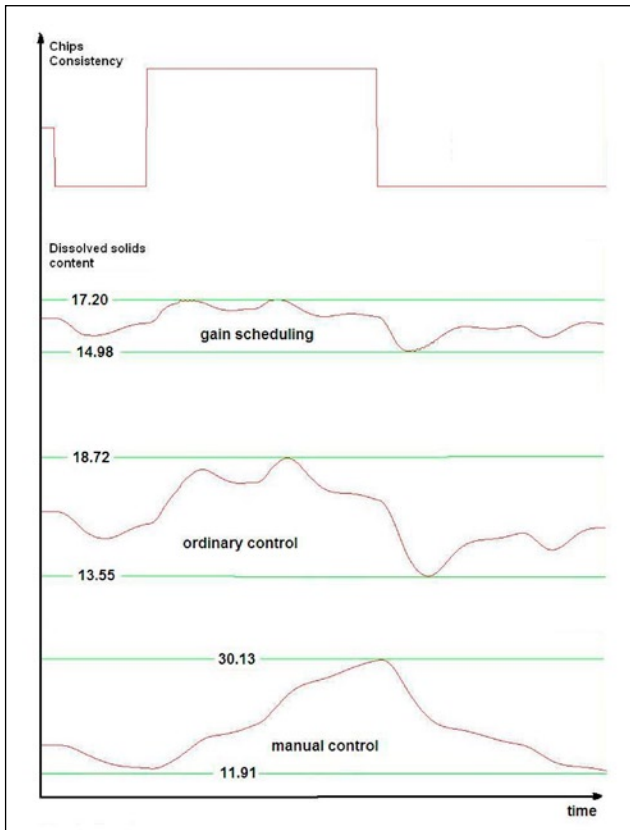
When the time constant is large, Skogestad's half rule method can be used to limit the value of the integral time τ_i [10], as in Eq. 5b:

$$\tau_i = \min \{ \tau, 4 (\lambda + \theta) \} = 240 \text{ min} \quad (5b)$$

For a rapid response of the PI controller τ_i was chosen at 180 min. The major problem of these fixed tuning parameters is the poor performance observed when load changes in chip consistency affect the process. A gain scheduling strategy



5. Dynamic response in dissolved solids content (top) after a bump on fresh water flow (bottom).



6. Comparing washed pulp dissolved solids content for three types of control, square perturbation.

would be more efficient.

In gain scheduling the controller uses higher gains when the error is greater and lower gains when the process value is closer to the set point. Eq. 6 shows how this is achieved [10]:

$$K_c = K_{c0} (1 + a |e(t)|) \quad (6)$$

The lambda factor can be calculated (Eq. 7) by combining Eq. 5 and Eq. 6:

$$\frac{1}{\lambda} = \frac{1}{\lambda_0} (1 + a |e(t)|) \quad (7)$$

where:

λ_0 = value of lambda factor when the error is zero ($\lambda_0 = 60$ minutes)

a = parameter fixed by the user to increase the speed of the controller.

The modified simulation works as follows:

- The user specifies the operating conditions of the mill;
- The Jacobian module computes the corresponding process gain K_p ;
- The user also specifies the parameters “ a ” and “ λ_0 ” in Eq. 7;
- The modified simulation computes the corresponding value of λ using Eq. 7;
- The modified simulation computes the controller gain K_c using Eq. 5;
- The new controller gain K_c is dynamically fed to the PI controller in Fig. 4.

Results from simulations

The following simulations compare the results between ordinary control (fixed tuning parameters), gain scheduling control, and manual control (constant flow value with no reset at major production changes, which still exists in some pulp mills [11]).

Chip consistency was allowed to swing in the operating interval of 45%–55% for one day of production (1440 min.). **Figure 6** shows the variability in dissolved solids content. Variability is lowest at 7.04% with gain scheduling, higher at 17.08% with fixed parameter control, and highest at 58.90% with manual control (**Table II**).

Another important result is that overall operating costs can be reduced. In comparison with ordinary control, the gain scheduling strategy offers a moderate reduction in operating costs, in the order of US\$ 20,440/year (US\$ 56/day). The reduction is more important in comparison with manual control (US\$ 103,660/year, US\$ 284/day). Each mill would have different costs and different optimal dilution factors, so the real operating cost reductions would be mill specific. **Table III** shows a comparison of operating costs for a square chip consistency perturbation signal.

Figure 7 shows clearly that the quantity of washing water used in manual mode is not sufficient. Gain scheduling uses

Control strategy	μ	σ	Variability	Maximum dissolved solids	Minimum dissolved solids
Ordinary control	16.22 kg/t	1.39 kg/t	17.08%	18.72 kg/t	13.55 kg/t
Manual control	18.84 kg/t	5.55 kg/t	58.90%	30.13 kg/t	11.91 kg/t
Gain scheduling	16.24 kg/t	0.57 kg/t	7.04%	17.20 kg/t	14.98 kg/t

II. Product quality comparison for a square chip consistency perturbation signal.

PROCESS CONTROL

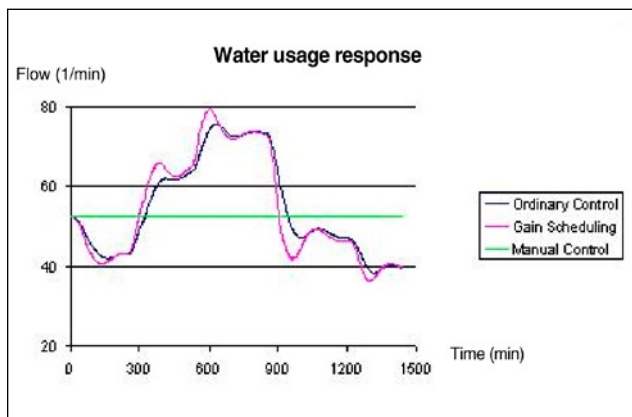
Control strategy	Operating costs (USD/day)	Costs (USD/t)
Ordinary control	11457	12.87
Manual control	11685	13.13
Gain scheduling	11401	12.81

III. Operating costs comparison for a square chip consistency perturbation signal.

20 tons less washing water per day than ordinary control strategy. When the consistency is raised from 45% to 55%, the gain scheduling strategy uses more water than ordinary control to maintain dissolved solids concentration near the set point, which is normal because the system now uses more chemicals. **Table IV** shows a comparison of washing water usage for a square chip consistency perturbation signal.

RGA APPLICATION

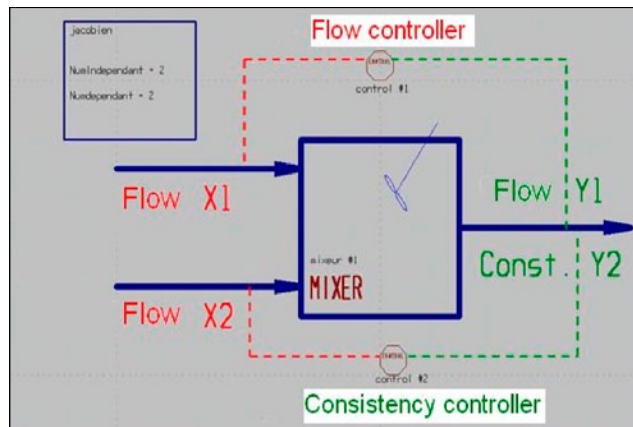
In most chemical processes, many variables must be manipulated and controlled. The controller's adjustments must take into consideration the effects of loop interactions that have a



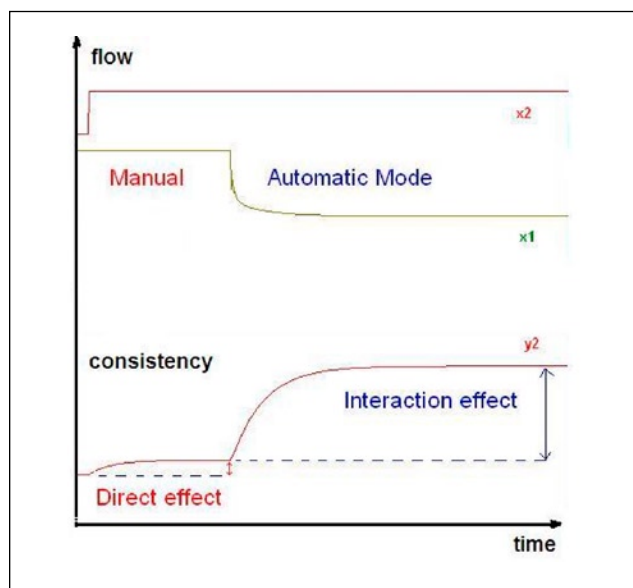
7. Washing water usage.

Control strategy	Mean water usage (l/s)	Total water usage (t/d)
Ordinary control	54.72	4731.5
Manual control	52.16	4506.6
Gain scheduling	54.49	4711.3

IV. Washing water usage comparison for a square chip consistency perturbation signal.



8. Multivariable control of a simple mixing system.



9. Interaction effect on the loop gain.

direct impact on stability and performance of the control system. The measure of the interaction (λ_{ij}) by the RGA is very useful to pair the manipulated and controlled variables to minimize the effect of interaction and adjust the controller gain K_c .

Figure 8 shows the control system of a blending tank where two flows x_1 and x_2 are manipulated to control the flow y_1 and the consistency y_2 of the outlet stream. In this example x_1 represents the dilute flow stream (2% consistency) and x_2 is the concentrated flow (70% consistency). Two PI controllers were used to reach the control goals. The flow controller manipulates the dilute flow stream x_1 . The consistency controller adjusts the concentrate flow x_2 . Lambda tuning method was used to calculate controller gains and integral times for the two controllers.

Loop interaction problem

Loop interaction occurs when a manipulated variable affects more than one controlled variable. In the blending control

system, the manipulated flow variables x_1 and x_2 affect both the outlet flow y_1 and consistency y_2 .

Figure 9 shows the response of the controlled variable y_2 after a change in the concentrated flow x_2 with flow controller in manual mode (direct effect) and the interaction effect caused by switching the flow controller to automatic mode. It is clear that the loop gain is more important when the flow loop is closed. From Fig. 9 we can compute the open loop gain J_{22} and the closed loop gain J'_{22} between the manipulated variable x_2 and controlled variable y_2 .

When the loops are opened, the loop consistency gain calculated after a 10% change in the concentrated flow x_2 is (Eq. 8):

$$J_{22} = \frac{\Delta x_2}{\Delta y_2} = \frac{60.78 - 60}{187.6 - 170.6} = 0.046 \text{ (\% / kg / s)} \quad (8)$$

The closed loop consistency gain K'_{22} calculated when the flow controller is in automatic mode is (Eq. 9):

$$J'_{22} = \frac{65.78 - 60}{187.6 - 170.6} = 0.34 \text{ (\% / kg / s)} \quad (9)$$

The relative gain is the ratio of the open loop and closed loop gains (Eq. 10):

$$\lambda_{22} = \frac{J_{22}}{J'_{22}} = 0.135 \quad (10)$$

This shows an important interaction since the closed loop gain is seven times greater than the open loop gain.

The Jacobian module computes the open loop gains J_{ij} , from which the RGA can be computed using Eq. 4. For the simulation of Fig. 8 the RGA matrix was computed as (Eq. 11):

$$RGA = \lambda_{ij} = \begin{matrix} & \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{pmatrix} 0.1388 & 0.8612 \\ 0.8612 & 0.1388 \end{pmatrix} \end{matrix} \quad (11)$$

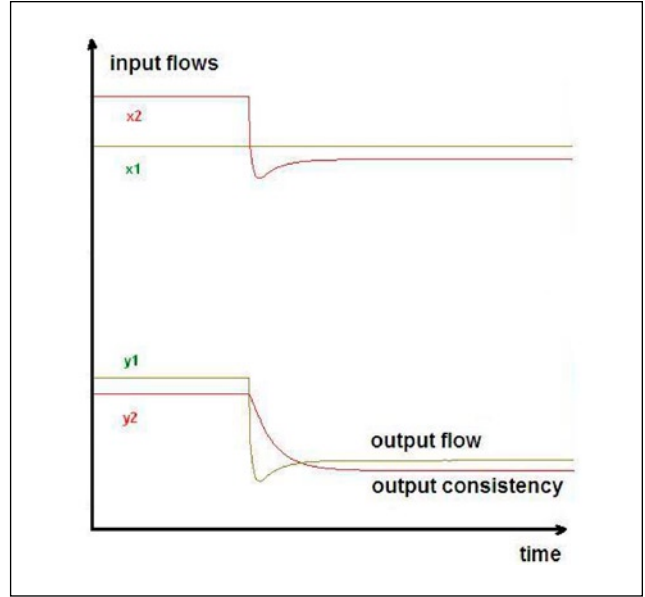
For a system with two manipulated and controlled variables, $\lambda_{22} = \lambda_{11}$ and $\lambda_{12} = \lambda_{21}$. Also, the sum of RGA line or columns elements is unity.

Controller design

The problem at hand is to find a pairing of manipulated-measured variables for the blending system. RGA calculations inform us that the pairings y_1-x_2 and y_2-x_1 are the best choice to minimize the interaction effect because λ_{12} and λ_{21} are closer to 1. Consequently, the open loop gain is approximately the same as the closed loop. On the other hand, the pairing y_1-x_1 and y_2-x_2 is not desirable for good control performance, due to the importance of the interaction ($\lambda_{12} = \lambda_{21} = 0.139$), which means that the closed loop gain is 7.2 times larger than the open loop gain, so this pairing must be avoided.

Control system performance

For the initial pairing y_1-x_1 and y_2-x_2 , two PI controllers, one for consistency and another for flow, were adjusted using the



10. Set point consistency change (60%–55%) with flow loop in manual, pairing y_1-x_1 and y_2-x_2

lambda tuning:

- PI consistency controller parameters: $K_c = 91.3$, $\tau_i = 21$ minutes; and
- PI flow controller parameters: $K_c = 0.33$, $\tau_i = 1$ minute.

Figure 10 shows good performance of the consistency controller when a set point change from 60% to 55% is made with flow controller in manual. But when the flow controller is switched to automatic, the control performance for the same set point change is poor (**Fig. 11**). Manipulated variable x_2 changes rapidly, which results in an overshoot of the controlled variable y_1 .

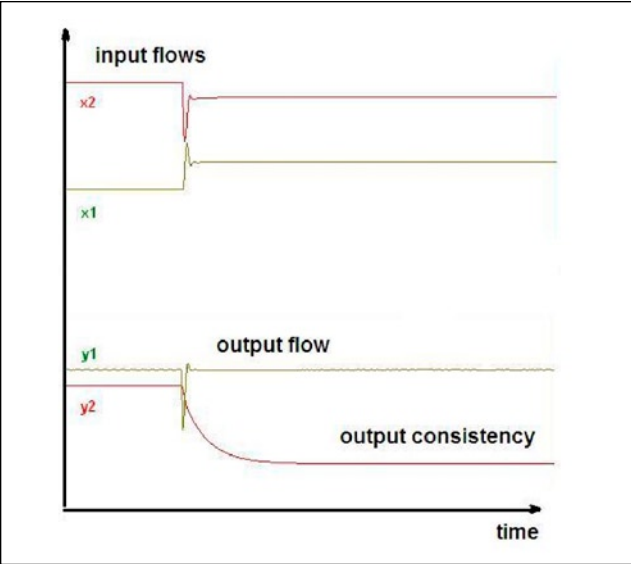
Figure 12 shows much better controller performance after the same set point change with both loops on automatic mode when the new pairing (y_1-x_2) and (y_2-x_1) is used instead. The controlled variable y_2 reached the set point without overshoot of the manipulated variables x_2 and x_1 , while flow y_1 remained practically unchanged.

Pairing choices at different operating points

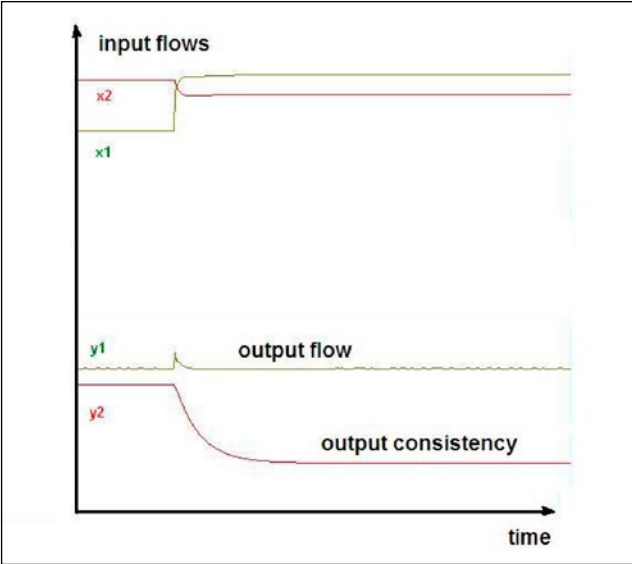
Process nonlinearities influence the value of the gain at different operating points. This leads to changes in the open loop gain matrix J and consequently to differences in the RGA matrix. For the blending system, which is a nonlinear process, the RGA matrix can be formulated symbolically as (Eq. 12):

$$\lambda_{ij} = \begin{matrix} & \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{matrix} y_1 \\ y_2 \end{matrix} & \begin{pmatrix} \lambda_{11} = \frac{x_1}{x_1 + x_2} & \lambda_{12} = \frac{x_2}{x_1 + x_2} \\ \lambda_{21} = \frac{x_2}{x_1 + x_2} & \lambda_{22} = \frac{x_1}{x_1 + x_2} \end{pmatrix} \end{matrix} \quad (12)$$

The relative gains are shown as functions of the operating



11. Set point consistency change (60%–55%) with flow loop in automatic, pairing y_1 - x_1 and y_2 - x_2 .



12. Set point consistency change (60–55%) with both loops on automatic, pairing y_1 - x_2 and y_2 - x_1 .

points, so the best pairing depends on operating conditions [12].

Table V confirms that open loop and relative gains calculated by the Jacobian module change with different operating points. Therefore, the designer should select the pairing of variables to minimize interaction effects and also analyze different “optimal” pairings found depending on the operating conditions. In this context the Jacobian module developed in simulation is a design tool that can help speed up the analysis of the different combinations of optimal pairings at different operating points.

CONCLUSIONS

We described possible simulation and control applications of a newly developed Jacobian module. The module performs automatic bumps of the manipulated variables and computes controller gains at different operating points in the form of a Jacobian matrix. The RGA matrix is also automatically obtained as a result.

In a simulated brown stock washing system, where washing efficiency is continuously perturbed by chip feed consistency, the Jacobian module was used to implement a gain scheduling strategy. More efficient washing, less variability,

and lower operating costs resulted in comparison with ordinary and manual control.

In the multivariable control example, the RGA matrix computed by the Jacobian module helped in the design of a control system that minimizes loop interactions in order to achieve the best control performance. The blending system example showed the utility of the Jacobian module in evaluating loop interactions as well as choosing pairings of variables that optimize control performance. The module also made it possible to analyze different optimal pairings at different operating points.

In its current version the Jacobian module performs simple single bump tests, the characteristics of which are user-defined. Future work would include the use of automated pseudo random bump sequences and automatic identification of the settling time. **TJ**

Received: September 18, 2007
Accepted: June 8, 2008

LITERATURE CITED

1. Sell, N.J., *Process control fundamentals for the pulp & paper industry*, TAPPI Press, Atlanta, Georgia, USA, 1995.

Operating conditions	Open loop gains	Relative gains	Pairing recommendation
$y_2 = 60\%$, $y_1 = 200$ kg/s	$K_{11} = 1$, $K_{12} = 1$ $K_{21} = -0.286$, $K_{22} = 0.046$	$\lambda_{21} = 0.86$ $\lambda_{22} = 0.14$	y_2 - x_1 and y_1 - x_2
$y_2 = 10\%$, $y_1 = 200$ kg/s	$K_{11} = 1$, $K_{12} = 1$ $K_{21} = -0.037$, $K_{22} = 0.297$	$\lambda_{21} = 0.11$ $\lambda_{22} = 0.89$	y_2 - x_2 and y_1 - x_1

V. Effect of operating conditions on pairing choices.

2. Bequette, B.W., *Process control: modeling, design, and simulation*, Prentice Hall, Upper Saddle River, New Jersey, USA, 2003.
3. Bialkowski, W.L., *Pulp & Paper* 72(12): 79 (1998).
4. Corripio, A.B., *Tuning of industrial control systems*, 2nd Edn., Instrument Society of America, Research Triangle Park, North Carolina, USA, 2001.
5. Gao, J. and Budman, H., *J. Process Control*, 15(7): 807 (2005).
6. Doyle, J., Kwatra, S., and Schwaber, S., *Chem. Eng. Sci.*, 53(15): 2675 (1998).
7. Edgar, T.F. and Lee, J. "Interaction measure for decentralized control of multivariable processes," *Proceedings of 2002 American Control Conference Vol. I*, American Automatic Control Council and IEEE, Anchorage, Alaska, p.454.
8. Wasik L., Nelson, D., and Mittet, G., *TAPPI J.*, 83(3): 94(2000).
9. Laperrière, L. and Wasik, L., *TAPPI J.*, 1(4): 7(2002).
10. Seborg, D.E., and Edgar, T.F., *Process dynamics and control*, 2nd Edn., John Wiley & Sons, Hoboken, New Jersey, USA, 2004.
11. Iverson, W., "Washing out energy costs," <http://www.automation-world.com/>, 2004.
12. Smith, C.A. and Corripio, A.B., *Principles and practice of automatic process control*, 2nd Edn., John Wiley & Sons, Hoboken, New Jersey, USA, 1997.

INSIGHTS FROM THE AUTHORS

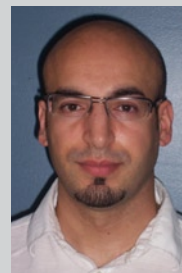
Process control is a great challenge for the industry. When measurements become more accurate and sophisticated, the PID controller must be optimal to respond accurately to process disturbances. Tuning parameters must be recalculated at different operating points to maintain the desired quality of the products and to minimize costs.

Gain scheduling and adaptive control have been widely researched and in our research group, these topics are part of a larger project on MOSOC (MOdeling, Simulation, Optimization, Control). The goal is to speed up new product/process development time by testing different designs in the virtual space. We used simulation to test controller design alternatives in the context of operational changes (gain scheduling) or in the context of new product/process control analysis (RGA).

Our challenge was to demonstrate the generic nature of the developed Jacobian module and the benefits that it could provide in very different simulation contexts. The Jacobian partials had to include process variables that were known to have an impact on key parameters such as cost, variability, or water usage. The values then had to be compared in control with and without activation of gain scheduling. We

were surprised by the difference that simple calculations can make in minimizing costs and tightening variability.

Mills could benefit from the use of this module if they simulate their process in parallel with real production conditions. Various controllers can be optimized by simple communication of the DCS (or the historian) with the simulator to achieve their objectives. Further work would introduce more advanced process bumping techniques like PRBS, as well as automated process identification of the main dynamic parameters.



Chbel



Laperrière

Chbel is a student and Laperrière is a professor at the Centre Intégré en Pâtes et Papiers (CIPP), Université du Québec à Trois-Rivières (UQTR), 3351 Des Forges Boulevard, Trois-Rivières, Québec, Canada G9A 5H7; Email Laperrière at Luc.Laperriere@uqtr.ca.