INTERACTIVE MODELING AND VISUALIZATION OF COMPLEX PROCESSES IN PULP AND PAPER MAKING

W. Birk[1], A. Johansson[1], M. Castaño[1], S. Rönnbäck[2], T. Nordin[3] and N.-O. Ekholm[4]

[1]) Department of Computer Science and Electrical Engineering, Luleå University of Technology, 97187 Luleå, SWEDEN
[2]) Optimation AB, Timmermansgatan 16, 972 33 Luleå, SWEDEN
[3]) SCA Obbola AB, Linjevägen 33, 913 42 Obbola, SWEDEN
[4]) Billerud Karlsborg AB, 952 83 Karlsborgsverken, SWEDEN

ABSTRACT

This paper discusses a new approach to interactive modeling, visualization and analysis of complex industrial processes. A theoretical framework based on signal flow graphs for modeling and visualization is presented. Using this framework a software tool is designed, called ProMoVis, which can be used to model a process, to visualize the models together with process construction and control system, and to perform analysis regarding *e.g.* feasible control strategies for the process. Moreover, a case study is conducted, where ProMoVis is used to model, visualize and analyze a stock preparation plant. The results indicate that the proposed methods and tools improve work flows, increase process understanding and simplify decision making on control strategies for complex process.

INTRODUCTION

The pulp and paper industry is operating in a very competitive and globalized environment, requiring efficient and updated production processes. Generally, processes are kept updated by maintenance or renewal of hardware and by refinement of the control strategies in the control system in order to cope with new production targets.

Obviously, the overall performance of the production process depends on good condition of the hardware and control strategies. Controllers need to be kept updated and well tuned for desired production targets and current hardware, and that, on different hierarchical levels.

Another complication arises due to the complexity of process industry plants, where hundreds or even thousands of variables are connected through dynamic systems. Examples of such interconnections are material flows and reflows, the latter *e.g.* due to discarded material being returned to previous process steps which gives rise to large feedback loops. Other examples are connections through supply grids for *e.g.* pressurized air. One process step consuming pressurized air may give rise to a pressure drop that propagates to every other consumer in the plant. Adding control loops to the process on both low and high level may result in a system with unintelligible causality and unpredictable dynamics.

For the control engineer, these very complex interconnected systems are a challenge. The question is how to represent and visualize the complexity in a comprehensible way and how to analyze it regarding *e.g.* control structure selection and dynamic behaviour. Software tools for visualization of complex system are *e.g.* ChemCAD, Matlab/Simulink, Extend and Dymola, the latter based on the generic modeling language Modelica. However, these are mainly intended for simulation and do not support the desired analysis tools for increased process understanding or decision making on changes of the control strategy. Additionally, there is usually a focus on the components or blocks of a complex system instead of variables and their interconnections.

As indicated in [1], visualization is important both from a collaborative perspective as well as to provide a comprehensive understanding of processes. Within the areas of construction, manufacturing or production management, visualization is recognized as an important tool, see [2], [3].

Experiences from collaboration with industry partners indicate that control engineers are in the need of tools that merge graphical representation of plant and control system with analysis results that indicate feasible control strategies. This would improve argumentation for changes and potential improvement of control performance. Moreover, the presented information should not be limited to the geographical closeness to where control action takes place.

A directed graph is a highly abstract way of visualizing complexity in various applications. By letting the nodes represent signals and the edges linear dynamic systems, one obtains the signal flow graph [4], which is a very general representation for interconnected, dynamic, linear systems. Compared to a block diagram, the signal flow graph has the advantage of being closely related to an algebraic representation, in terms of matrices consisting of the edges in the graph. Similarly, the signal flow graph allows choosing the level of detail in the representation, thus encompassing both the input/output form as well as the state-space form as special cases. Tools for decomposing interconnected systems in state-space form are considered in *e.g.* [5], [6] and another special case is treated in [7] where an autoregressive (AR) structure of the edges is assumed.

The aim of this paper is to present a theoretical framework and a software tool for visualization and analysis of complex dynamic processes. The paper is arranged as follows. First the theoretical framework is described, followed by a discussion of the prototype tool that implements the framework. Thereafter a case study is presented where the tool is used to model, visualize and analyze the stock preparation plant at SCA Obbola AB. Finally, some conclusions are given.

VISUALIZATION USING THE SIGNAL FLOW GRAPH

To represent interconnected systems, we will use the signal-flow graph which consists of nodes that represent the signals and edges that describe how the signals affect each other, see Fig. 1. Three categories of nodes are considered, input signals u_i, $i = 1..p$, output signals y_i, $i = 1..q$, and internal signals z_i, $i = 1..n$. The interpretation of these categories may vary depending on the problem at hand, *e.g.* if the signal flow graph

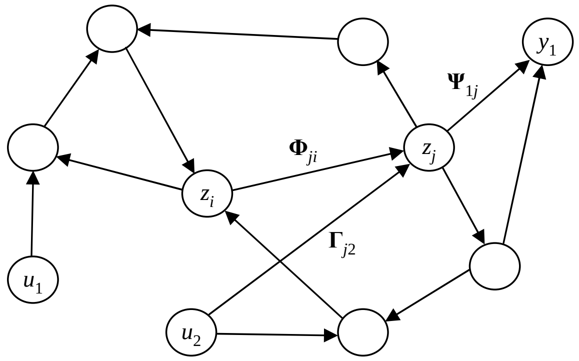


Figure 1. A graph for representing an interconnected system. The nodes are signals and the edges are linear, dynamic systems

is supposed to represent only the process or a closed loop system including controllers. The internal signals are assumed to be affected linearly by the input signals and the other internal signals, i.e.

$$z_i = \mathbf{\Phi}_{i1} z_1 + ... + \mathbf{\Phi}_{in} z_n + \mathbf{\Gamma}_{i1} u_1 + ... + \mathbf{\Gamma}_{ip} u_p \quad (1)$$

for $i = 1..n$ where $\mathbf{\Phi}_{ij}$ and $\mathbf{\Gamma}_{ij}$ are linear dynamic systems. Similarly, the output signals are assumed to be affected linearly by the internal signals and the inputs as

$$y_i = \mathbf{\Psi}_{i1} z_1 + ... + \mathbf{\Psi}_{in} z_n + \mathbf{\Omega}_{i1} u_1 + ... + \mathbf{\Omega}_{ip} u_p \quad (2)$$

for $i = 1..q$. By collecting the signals u_i, z_i., and y_i into vectors u, z, and y and defining the multivariable, dynamic systems $\mathbf{\Phi}$, $\mathbf{\Gamma}$, $\mathbf{\Psi}$, and $\mathbf{\Omega}$ whose i, jth element are $\mathbf{\Phi}_{ij}$, $\mathbf{\Gamma}_{ij}$, $\mathbf{\Psi}_{ij}$, and $\mathbf{\Omega}_{ij}$, respectively, the signal flow graph representation may now be formulated as

$$z = \mathbf{\Phi} z + \mathbf{\Gamma} u \quad (3a)$$

$$y = \mathbf{\Psi} z + \mathbf{\Omega} u \quad (3b)$$

It is straightforward to see that the internal variables are related to the inputs as $z = (I - \mathbf{\Phi})^{-1} \mathbf{\Gamma} u$ and that the input/output relation is

$$y = (\mathbf{\Psi}(I - \mathbf{\Phi})^{-1} \mathbf{\Gamma} + \mathbf{\Omega}) u$$

We may now introduce the concept of visualization from a mathematical point of view, by which we mean the representation of a linear system to show "internal wirings" in the form of a signal-flow graph. Given a linear, multivariable system $\mathbf{G}$, the quadruple $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{\Psi}, \mathbf{\Omega})$ is called a *visualization* of $\mathbf{G}$ if $\mathbf{\Psi}(I - \mathbf{\Phi})^{-1} \mathbf{\Gamma} + \mathbf{\Omega} = \mathbf{G}$. The kinship to the concept of state-space realization should be clear. Indeed, given a realization of the system $\mathbf{G}$ as $G(s) = C(sI - A)^{-1} B + D$, a visualization with the internal variables z chosen as the state variables of the realization is given by $\Phi(s) = A/s$, $\Gamma(s) = B/s$, $\Psi(s) = C$, and $\Omega(s) = D$. A realization can thus be considered as a special case of a visualization, with all state variables visible as internal variables. Another special case is when no internal variables are visible, and is obtained by the trivial choice $\Omega(s) = G(s)$ and $\Gamma(s)$, $\Phi(s)$, $\Psi(s)$ equal to zero.

Example 1: Consider the simple example of a double water tank process as illustrated in Fig. 2(a) where the control signal to the pump is the input u while the tank levels are the internal variables z_1 and z_2, respectively. The process is assumed to be modelled by

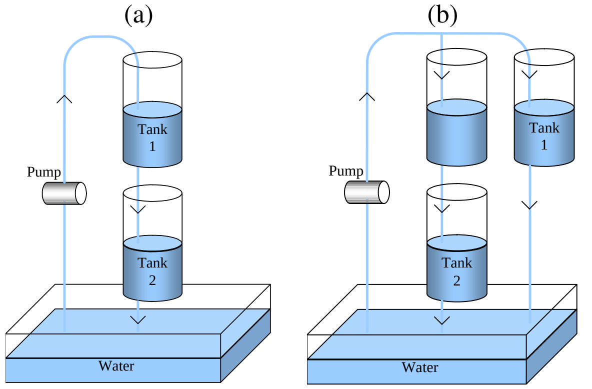


Figure 2. (a) The water tank process considered in Example 1. (b) another physical system with the same values of the internal variables

$$z_1(s) = \frac{k}{s\tau + 1} u(s) \quad (4)$$

$$z_2(s) = \frac{1}{s\tau + 1} z_1(s) \quad (5)$$

for some constants $k, \tau > 0$, *i.e.* as a visualization $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{\Psi}, \mathbf{\Omega})$ with

$$\Phi(s) = \begin{bmatrix} 0 & 0 \\ \frac{1}{s\tau+1} & 0 \end{bmatrix}, \quad \Gamma(s) = \begin{bmatrix} \frac{k}{s\tau+1} \\ 0 \end{bmatrix} \quad (6)$$

and $\Psi(s), \Omega(s)$ arbitrary since we are not concerned with the outputs in this example. Equivalently, (4) may be substituted into (5) so that

$$z_1(s) = \frac{k}{s\tau + 1} u(s)$$

$$z_2(s) = \frac{k}{(s\tau + 1)^2} u(s)$$

i.e. the same response of the internal variables to the input but with the visualization $(\mathbf{\Phi}', \mathbf{\Gamma}', \mathbf{\Psi}, \mathbf{\Omega})$ with

$$\Phi'(s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \Gamma'(s) = \begin{bmatrix} \frac{k}{s\tau+1} \\ \frac{k}{(s\tau+1)^2} \end{bmatrix} \quad (7)$$

which rather corresponds to the tank setup in Fig. 2(b).

The important point that is stressed with this example is that although visualizations are nonunique, some visualizations may be consistent with the process physics while others are not. The two visualizations in Example 1 have different physical structure in *e.g.* the sense that a perturbation introduced in z_1 affects the variable z_2 in the visualization (6) but not in (7).

Operations on signal flow graphs

Hiding of nodes

When visualizing an interconnected system, one critical issue is the ability to disregard unnecessary details. A complex industry facility is likely to include hierarchies of controllers, *e.g.* cascade control structures. In a graphical representation of a very complex plant, we would thus like to be able to choose the level of abstraction so that when dealing with supervisory control loops, internal variables of low level loops can be hidden. Thus, a central operation is the hiding of a set of nodes that are not presently important, without changing the relation between the inputs and the remaining internal variables and without changing the physical structure (which happens in Example 1). In [8] a hiding operation that preserves the physical structure of the interconnected system is defined and it is also shown that this hiding operation is commutative, *i.e.* that hiding of a set of nodes can be performed in any order.

Eliminating self references

A node that depends on itself (*i.e.* a nonzero diagonal element in $\mathbf{\Phi}$) may be disadvantageous for the visual interpretation of the interconnected system. Thus, in [8], an operation that eliminates edges from a node to itself is defined. Similar to the hiding operation, this elimination operation is commutative and preserves the relation from the inputs to the internal variables. Furthermore, it commutes with the hiding operation in a certain sense, so that these operations can be performed in arbitrary order.

Analysis tools for signal flow graphs

The signal flow graph is a very general framework on which many different analysis tools can be defined. Examples are tools for selecting control structures and for determining which edges are important in different regards. The importance of an edge can support decisions on whether it can be replaced by an approximation or even neglected. Another important application is *clustering* of the interconnected system [5], [6], *i.e.* grouping nodes that have strong influence on each other (as opposed to grouping nodes that are geographically close).

Brain connectivity based measures

A basic measure is an operator norm taken on each edge (termed norm index) but, as shown in examples in [9], this quantity does not give deep insight into the properties of the system. An improvement is to weigh this quantity relative to the sum of the norms of all edges entering or leaving a node, which gives rise to the brain connectivity based measures in [7]. These measures can also be applied frequency-wise resulting in a function of frequency instead of a scalar number for each edge. Some of them apply to the input/output system, *i.e.* where all internal variables are hidden.

Loop index

It is well known that closing a loop may alter the behavior of a dynamic system, particularly if the loop gain (with respect to some signal norm) is large. Indeed, if the gain is larger than unity then instability may occur. In [8] the loop index is thus introduced, which measures the gain of the loop that an edge closes.

Controllability index

Another interesting property of an edge is how much it affects the controllability of the system. Controllability is here defined as the minimum energy for bringing the interconnected system from the origin to a given steady state [9].

Control structure design tools

Typical methods from control structure design can be used on signal flow graphs and are then applied to the input/output representation. These methods comprise interaction measures that are based on relative gains or system gramians, [10], [11], [12]. Additionally, methods that target the reconfiguration of control strategies based on the current ones are also possible to apply, [13].

MODELING AND VISUALIZATION TOOL

Making use of the signal flow graph framework a prototype for a computerized tool called ProMoVis (Process Modelling and Visualization) is now created. Its purpose is to give a graphical visualization of both plant and control system, and to perform process analysis and display the results in a comprehensive way. The targeted users are control engineers working with the design and improvement of control strategies in process industry plants.

The tool has been created from the perspective of immediate usability in industry which means that user choices were limited to direct the user in their work with the tool. Thereby, the getting started time is reduced for new users but advanced users might feel the limitations as disturbing.

Generic objects

In ProMoVis, there are four (4) types of objects: Variables, process models, controllers and components. The variables represent the signals (nodes) in the signal flow graph framework, which can be grouped into categories based on their principal character: measured or controlled variable, reference variable, manipulated variable, disturbance variable, estimated variable, state variable. These groups are of importance as they determine how variables can be interconnected and when they are visible. Usually, estimated variables are only present in the control system, whereas disturbance variables and state variables are only present in the process. Measured and manipulated variables are present in both process and control system, and represent the interface between process and control system. The choice of which variables are collected into the vectors u, z and y depends on the analysis that is performed. When hierarchies are considered, *i.e.* in cascade control, not only the process becomes subject to the analysis but also parts of the control system. In the current version of ProMoVis this is not possible in order to simplify the usage and implementation, but in future releases controller hierarchies need to be considered.

The process models correspond to the edges of the signal flow graph and are the interconnections between variables representing the dynamic behaviour of the plant. Principally, process models can be defined on a single-input-single-output basis, but multi-input-multi-output models are supported as well. Additionally, a process model can be either defined in continuous or discrete time. In order to simplify modeling efforts, some process model structures which are used within system identification are pre-defined. The actual parameter identification for models is not implemented in the tool, as sufficiently well developed tools are already available.

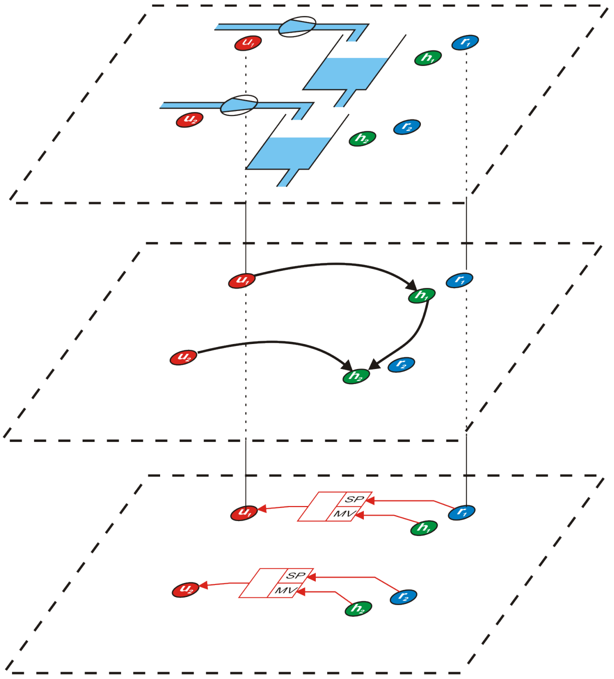


Figure 3. Different layers in the modeling and visualization concept. Process layer (top), Process models (middle), Controllers (bottom). Manipulated variables (red), Measured or controlled variables (green), Reference variables (blue).

Additionally, controllers are defined, which principally do not differ from process models in their implementation and could be depicted by two edges, from reference and controlled variable to manipulated variable. Instead controllers are displayed as blocks with two inputs (reference variable, controlled variable) and one output (manipulated variable), as shown in the lower part of Fig. 3. The reason is to simplify for user to create and connect controllers properly and to avoid mistakes. Similar to process models, some controllers types are pre-defined, like *e.g.* PID controllers.

Components are used to graphically represent the plant, and have no other functionality than providing a geographic and constructive understanding of the plant with a rather coarse level of detail and realism. By using symbols according to the SSG standard or bitmap images of drawings or sketches it is possible to create a simplistic but effective representation.

Non-linearities are usually present in industry processes. Since the framework and most analysis methods require linear dynamic system representations, linearized versions of the system dynamics around working points have to be used. Thus, working points become a property of the variables. This means that we may have different process models for different working points.

In other words, working points could be interpreted as variants of the complete system, which means that the visualization of a plant will come with a set of variants. Each variant may have different process models, variable properties and controllers. Another cause for the generation of a variant would be the assessment of different control strategies for one and the same working point, which differs from the latter reason. It also needs to be noted that variants should not be mistaken for versions which are merely the evolution of a visualization over time.

Layering and views

When a process plant is represented in ProMoVis, the underlying representation is a signal flow graph as exemplified in Fig. 1. In contrast, the user understands the signal flow graph more in the structure which is displayed in Fig. 3. Clearly, this view is more adapted to the way a control engineer understands a process plant which is composed of different kinds of elements as defined in the previous subsection.

From a visual perspective it also becomes very complex to have all elements visible at the same time, which is of interest during composition or building, but unadvisable during analysis and discussion. In latter case it is of interest to select certain types of information that should be viewed. This can be understood as layering. It is a design choice of the tool to make process models, components and variables to reside in one layer each and that users can group controllers in additional layers according to their preference.

From an analysis perspective it is also important to note that the placement of variables according to geographical location is important during the composition of the model what may need to be changed during analysis *e.g.* to group variables that have a significant effect on each other. This type of abstraction is a necessity to make good decisions on control strategies, as it removes geographical location of variables as a factor in the decision making.

Analysis of processes

When the user has composed a process plant in ProMoVis and introduced all the necessary parameters for variables, process models and controllers, it is possible to analyze the process to design a control strategy. When an analysis is performed, first a subset of variables is chosen for which the analysis is performed. Thereafter, different tools can be accessed and the tool performs the necessary mathematical computations. All computations are run in symbolic notation in order to reduce numerical errors. The result of the analysis can have different character: An array of numbers, a directed graph representing the significance of the interconnection in the line width of the edges or an array of frequency dependent plots that shows the significance of interconnection depending on the frequency of the exciting signals.

Examples of analysis results are given in the following case study.

CASE STUDY: A STOCK PREPARATION PLANT

The stock preparation plant is a subprocess present in any paper mill. The refining section of the stock preparation plant in SCA Obbola AB is here considered. In conventional refining, the pulp is pumped through the gap between two coaxial grooved discs. A moving disc can be rotated and displaced in the axial direction, and the friction of the fibres with the discs and with each other creates the refining effects. Refining creates major changes in pulp properties. External fibrilation is the most desired of the effects, improving the fibre bonds at the forming section. Refining also creates undesirable effects on the pulp, i.e. internal fibrilation has a large impact in the dewatering capacity of the paper web,

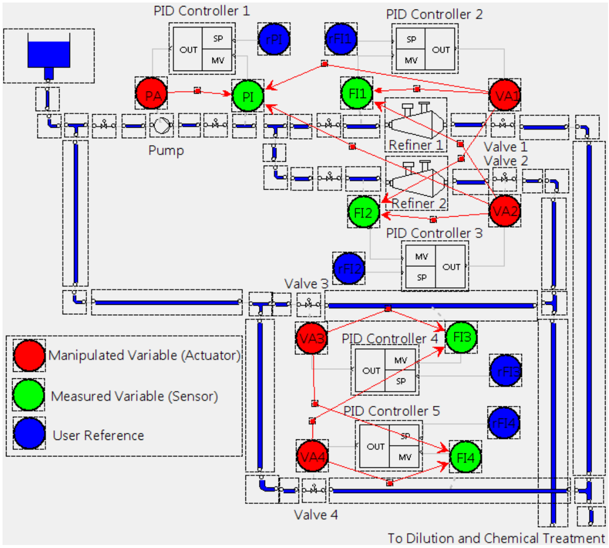


Figure 4. ProMoVis screenshot. Refining section of the stock preparation plant at SCA Obbola.

Table 1. Considered sensors and actuators in the refining section.

Actuators

Tag	Name	Description
PA	Pump Actuator	Pumps the flow through the refiners
VA1	Valve Actuator 1	Valve after refiner 1
VA2	Valve Actuator 2	Valve after refiner 2
VA3	Valve Actuator 3	Valve at the recirculation from refiner 2
VA4	Valve Actuator 4	Valve at the recirculation from refiner 1

Sensors

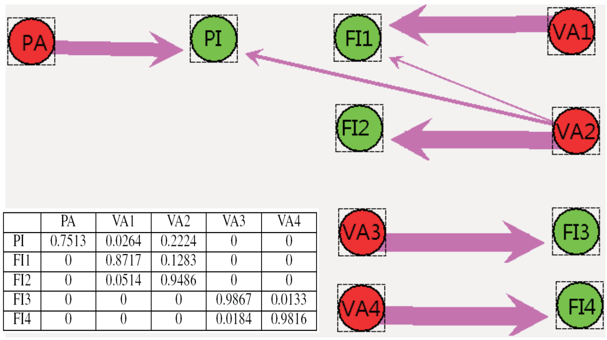
Tag	Name	Description
PI	Pressure Indicator	Pressure before the flow bifurcation
FI1	Flow Indicator 1	Pulp flow though refiner 1
FI2	Flow Indicator 2	Pulp flow through refiner 2
FI3	Flow Indicator 3	Pulp flow recirculated from refiner 2
FI4	Flow Indicator 4	Pulp flow recirculated from refiner 1

short fibre flocs may have a negative impact in the paper forming, and the formation of a large amount of fines have to be avoided, since they have to be retained by the paper web at the wire section.

For the refining section of the stock preparation plant in SCA Obbola, the process and the existing controller are depicted in Fig. 4. First the pulp is pumped from a storage tank and the flow bifurcates towards two parallel refiners. Note that a fraction of the pulp is recirculated again for a finer refining. This recirculation increases the complexity of the process, requiring a deep analysis of the process interconnections in order to understand the process and design a control structure.

The set of considered sensors and actuators is summarized in table 1. The refiners have internal controllers to track a setpoint for the energy delivered to the pulp. Safety, quality, and production depend on well maintained setpoints for the considered flows and the pressure at the entrance of the refiners. In the current control of the process, four independent scalar PID controllers are used to maintain the flows at the desired setpoints. The centrifugal pump is then used as actuator in another control loop to keep the pressure before the refiners constant.

The structure which results from closing scalar controllers by selecting pairs of sensors and actuators is known as *decentralized control* structure. This technique is very popular due to its simplicity and easy maintenance. Nevertheless, loop interactions often result in oscillations in the control loops, which can be reduced by increasing the complexity of the control structure.



	PA	VA1	VA2	VA3	VA4
PI	0.7513	0.0264	0.2224	0	0
FI1	0	0.8717	0.1283	0	0
FI2	0	0.0514	0.9486	0	0
FI3	0	0	0	0.9867	0.0133
FI4	0	0	0	0.0184	0.9816

Figure 5. ProMoVis screenshot. Analysis of the stock preparation plant with the method FET_r. Either a graph, or the connectivity matrix related to the graph can be chosen as displayed result. The layers including the components, the process models, and the controllers with their corresponding references are selected as not displayed.

Implementation of the stock preparation plant in ProMoVis

The visual representation resulting from implementing the stock preparation plant in ProMoVis is depicted in Fig. 4.

First, a visualization of the physical layout of the process was created by connecting components representing elements as pipes, valves, pumps and refiners. Then a subset of sensors and actuators to be considered for control was selected, and the corresponding variables were defined in the visualization.

In order to collect significant process data for the modeling task, the process was excited during normal operation by perturbing the actuators with additive white nose. In a first modeling step, a model structure was created by identifying which actuators generate an observable impact on which measured variables. The actuator-sensor relationships corresponding to this model structure were modeled as dynamic models using system identification techniques. Each of the obtained actuator-sensor models was implemented in ProMoVis and is represented by a red arrow in Fig. 4.

Finally, the controllers representing the current control of the process were defined in order to visualize and maintain the information on the current control.

Analysis of the stock preparation plant with ProMoVis

FET_r, a tool based on brain connectivity, was applied to the ProMoVis model of the stock preparation plant, and the result is depicted in Fig. 5. In FET_r, the significance (width) of all the edges entering a measured variable add up to one, and they represent the relative effect of the process actuators. The most significant edges entering a measured variable identify the actuators which can deliver a higher energy contribution on the measured variable. An optional threshold on the significance of the edges to be displayed was placed at 0.1, simplifying in this case the analysis by neglecting the edges which are considered to be insignificant.

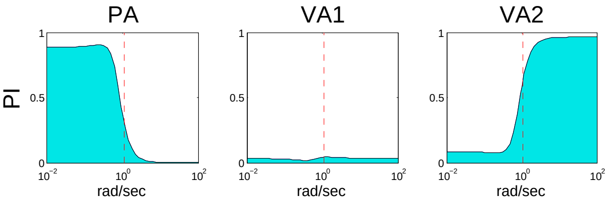


Figure 6. ProMoVis screenshot. The tool $FDPT_r$ describes the contributions on PI from the actuators in the frequency domain. The largest crossover frequency of all the considered actuator-sensor channels is marked by a dashed line.

A controlled variable should be associated with the minimum number of actuators that result in a large enough value of the sum of their contributions (edge widths). In general, there is no theory stating which one is the minimum sum of contributions wthat gives an acceptable control structure. However, based on the experience with similar interaction measures [11], we will assume consider that a value larger than 0.7 should be achieved to be on the safe side.

By inspecting Fig. 5, and pairing each of the measured variables with the actuator connected with the most significant edge, it is clear that, the best decentralized control structure is the one already in use in the process, and it is expected to achieve a satisfactory performance It is also clear at first sight, that both recirculation branches can be assumed to behave as independent scalar subprocesses, and simple PID controllers can be independently designed for them without loop interaction. Nevertheless, it is suspected that there exist a potential of improving the control performance by considering the dynamic connection from VA2 to PI in the control system, since this will increase the sum of contributions on PI from 0.7153 to 0.9737.

To obtain a deeper insight on the effects on PI, the brain connectivity tool $FDPT_r$ is applied using ProMoVis, and the result is depicted in Fig. 6. This tool is a frequency domain description of the relative power contribution of the actuators on a given measured variable. At each frequency, the sum of all the contributions on a measured variable add up to one. It can be observed that the contribution from VA2 has an important impact at frequencies around the maximum crossover frequency of the considered channels, causing interaction between the control loops which may be translated into oscillations. This conclusion is supported by the fact that the centrifugal pump has rotor dynamics which are slower than the dynamics of the valve, and by the observations of the plant operators and engineers, which confirm the existence of the mentioned oscillations.

A potential of improving the existing control structure has therefore been identified. The suggestion is to consider the actuator-sensor connection from VA2 to PI in the control structure, i.e. with a feed-forward action.

CONCLUSIONS

In this paper we have presented a new theoretical framework for the representation of complex processes using signal flow graphs. The framework is used as the basis for a visualization tool ProMoVis that supports control engineers in the modeling and analysis task of their work by combining graphical representations of the plant, process models and controllers. Additionally, analysis results can be depicted in the current view of the model, *e.g.* as a directed graph with significance levels. The user has the possibility to choose which information is visible at the same time.

Moreover, a case study is presented where a stock preparation plant is modeled and analyzed using ProMoVis. In the analysis several interaction measures are used together with coherence functions. The results from the analysis using coherence functions are displayed as a directed graphs which indicate the significance of process models as thicknesses of the edges in the graph and as frequency plots displaying the significance over a frequency range.

It is shown that ProMoVis can be successfully used to interactively model, visualize and analyze complex processes. Still, ProMoVis has in its current implementation certain limitations that are targeted to be removed in future version of the tool. Some of the limitation require further research efforts, like *e.g.* the analysis of processes with time delays or non-linearities.

ACKNOWLEDGEMENT

The work has been conducted within the MeSTA project that is hosted at ProcessIT Innovations at Luleå University of Technology and run within the branch framework SCOPE. Funding provided by VINNOVA, Hjalmar Lundbohm Research Centre and the participants of the SCOPE consortium, is hereby gratefully acknowledged.

REFERENCES

[1] M. W. Rohrer, "Seeing is believing: The importance of visualization in manufacturing simulation," in *Proceedings of the 2000 Winter Simulation Conference*, J. Joines, R. Barton, K. Kang, and P. A. Fishwick, Eds., 2000.

[2] D. Bouchlaghem, H. Shang, J. Whyte, and A. Ganah, "Visualisation in architecture, engineering and construction (AEC)," *Automation in Construction*, vol. 14, pp. 287–295, 2005.

[3] T. R. Browning and R. V. Ramasesh, "A survey of activity network-based process models for managing product development projects," *Production and Operations Management*, vol. 16, no. 2, pp. 217–240, 2007.

[4] S. J. Mason, "Feedback theory - Some properties of signal flow graphs," *Proceedings of the IRE*, pp. 1144–1156, September 1953.

[5] M. E. Sezer and D. D. Šiljak, "Nested ε-decompositions and clustering of complex systems," *Automatica*, vol. 22, no. 3, pp. 321–331, 1986.

[6] A. I. Zečević and D. D. Šiljak, "A decomposition-based control strategy for large, sparse dynamic systems," *Mathematical problems in engineering*, no. 1, pp. 33–48, 2005.

[7] M. Castaño and W. Birk, "New methods for structural and functional analysis of complex processes," in *Proc of 2009 IEEE Multiconference on Systems and Control, St. Petersburg, Russia*, 2009, pp. 487 – 494.

[8] A. Johansson, "Commuting operations on signal-flow graphs for visualization of interconnected systems," 2010, to be presented at the *4th IFAC Symposium on System, Structure and Control* in Ancona, Italy, Sept. 2010.

[9] A. Johansson, "On signal-flow graphs for visualization and analysis of complex systems," in *Reglermöte 2010*, 2010.

[10] M. van de Wal and B. de Jager, "A review of methods for input/output selection," *Automatica*, vol. 37, pp. 487–510, 2001.

[11] M. E. Salgado and A. Conley, "MIMO interaction measure and controller structure selection," *International Journal of Control*, vol. 77, no. 4, pp. 367–383, 2004.

[12] W. Birk and A. Medvedev, "A note on gramian-based interaction measures," in *Proc. of European Control Conference ECC, Cambridge, UK*, 2003, pp. 2660–5.

[13] W. Birk, "Towards incremental control structure improvement," in *Proc of 46th IEEE Conference on Decision and Control, New Orleans, December 2007*, 2007.