ON-LINE PULP MILL PRODUCTION OPTIMIZATION

Jens Pettersson *,[1] **Ulf Persson** **
Thomas Lindberg * **Lars Ledung** * **Xiaojing Zhang** *

* *ABB Corporate Research, Västerås, Sweden*
** *ABB Automation Technologies, Västerås, Sweden*

Abstract: We consider the planning of pulp production for large sulphate and sulfite mills. The production planning problem is formulated as a non-linear program (NLP) given a process model of the mill as constraint. The objective is to minimize the usage of expensive chemicals and to minimize the (squared) deviation from specified set-points for selected variables, e.g. production, tank-level and chemical composition of the cooking liquor. The problem formulation also considers upper and lower limits on variables and limitations in the derivative of production related variables. The NLP, which involves several tens of thousands of variables, is solved using algorithms for large-scale optimization. To provide a correct initial state of the process model, a moving horizon estimation is done to estimate the current state of the process.
A model library consisting of common process units in pulp mills have been developed. The models are described by differential algebraic equations. A software platform, which enables the user to assemble complex process models of complete mills based on the model library, has been developed. The platform also serves as data collector for the measured values from process sensors, as well as storing optimized and estimated values.
The pulp mill production planning system is installed on-line at Billerud Gruvön, a large Swedish integrated pulp and paper mill, producing some 660000 tons of sulphate and sulfite pulp.

Keywords: Process control, optimal control, state estimation, non-linear programming

1. INTRODUCTION

The work presented in this paper was carried out jointly by ABB Corporate Research and ABB Automation Technologies in connection with the installation of an advanced production planning tool in the Billerud Gruvön Pulp mill. The aim was twofold: to solve some specific production planning and chemical composition control problems associated with the Gruvön mill. Secondary aims where to develop methods, procedures and tools for the efficient solution of the generic problem of controlling and optimizing large-scale complicated chemical processes described by differential-algebraic equations.

A modern pulp and paper plant is usually a very complicated process, characterized by a lot of recirculations of material. This is mostly an effect of the natural aim in reducing cost be re-using expensive chemical but also an effect of current environmental legislation, which have

[1] Corresponding author: jens.pettersson@se.abb.com

driven the mills to a high degree of closeness, where almost the only effluents are purified water and the pulp and paper produced in the mill. Also in Scandinavia, where most of the larger mills have an operating history dating back to the early 20th century, many mills have been gradually rebuilt and extended over the years which have created very complicated process configurations and layouts.

1.1 Pilot Mill

The pilot mill, Billerud Gruvön, is one of Sweden's largest pulp and paper producers with an annual production of some 660000 tons paper. The mill operates three pulp production lines with a common chemical recovery island. The pulp is consumed by six paper machines and two drying machines, respectively. Key issues for the production planning is to supply the paper machines with correct amounts of pulp while keeping the relation between sodium and sulfur in the cooking liquors at a fixed ratio. It is also necessary to keep track of various storage towers and buffers to prevent over- or underflows. Currently this is the task of the production engineers who manually decides the production in the mill but there is a clear need and wish to automate the planning process.

1.2 Proposed solution

Our approach to solve the planning problem is outlined in the following steps:

(1) Develop a mathematical model of the complete process based on relevant mechanism and reactions. The model should be able to adequately simulate and predict the mass- and chemical balance of the mill.
(2) Find a suitable optimization criteria where different objectives and requirements can be addressed.
(3) Use the model and the optimization criteria in a model-predictive framework where optimal production trajectories are calculated by solving the optimization formulation with the model as constraint. It is also desirable to have limits on some of the production related variables and their derivative.

This approach is clearly influenced by the concept of model predictive control (MPC), which over the last decade has gained an strong position within the process control community. However, it is not the primary intention to use this tool in a closed loop fashion. Instead we consider the usage of this tool as a scientific methodology and aid for the daily planning of a large pulp mill.

2. MODELING

Modeling plays a key role in being able to control the production and chemical composition of the plant. Without a suitable mathematical description of the process it is naturally no use in trying to optimize anything.

When designing mathematical models describing the material- and chemical balance of chemical process units it is quite natural to use differential algebraic equations (DAE). The dynamic equations comes typically from the mixing of components in large vessels and tanks. The algebraic equations arises from e.g. reactions and flow-balances, where the dynamics are very fast and thus negligible. As a notation we have used the following semi-explicit DAE:

$$0 = g(x(t), z(t), u(t), p) \tag{1}$$

$$\dot{x}(t) = f(x(t), z(t), u(t), p) \tag{2}$$

$$y(t) = h(x(t), z(t), u(t), p) \tag{3}$$

where x is a vector of state variables, z are the algebraic variables, u are the manipulated variables and y are the measured variables, respectively. p is a vector of time-invariant parameters, which are assumed to be constant over the calculation horizon but may vary during a longer time span (e.g. due to seasonal variations). The functions $g(\cdot,\cdot,\cdot,\cdot)$, $f(\cdot,\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot,\cdot,\cdot)$ are vector valued functions of appropriate dimension. For each object type, a relevant set of equations like 1-3 is defined. To create a composite model based on instances of different model types, the instances are connected by appropriate streams objects. This results in a similar system of equations, but at a larger scale.

2.1 Modeling tools

To handle the model development a set of Matlab-based tools was developed. The models of each object type (e.g. digester, pulp storage tower etc) are written and stored in separate text-files, using a very simple syntax. The syntax consists of a header with declarations of variables and a body where the equations are specified. Given a set of such files, the Matlab tool parses the files and makes a quick analysis of the syntax and definitions. Obvious error are detected. A tree of Matlab structs are created that represents the attributes of each object (e.g. connectors, equations, parameters etc).

As a first step, a Modelica library is created based on the model set. This library contains different packages of model classes, corresponding to the object types that are used in the on-line application. Using the Modelica based simulation environment Dymola, a two-level hierarchic model of the Gruvön mill was built. The bottom level in this hierarchy consists of separate models of the process sections (fiber lines, evaporator plant, etc) and the top level is the complete mill, in which the process section models are components together with their interconnecting streams and buffer tanks. These models were used to off-line simulate and analyze the process models. As a heuristic verification procedure the simulated response of the process models at different operating points was manually compared to operating data from the Gruvön mill. When the dynamic response, considering both amplitude and time-constants, was sufficiently close to the available measurements, a model was considered as verified. Typical tuning parameters where the model parameters p and initial values of the state variables. No more formal or quantitative method was used.

The Matlab tool also automatically generates the necessary model information to use in the on-line optimization and state-estimation steps, see Section 5.

3. OPTIMIZATION

To achieve the objectives with the production planning tool, a suitable objective function is necessary. To standardize the implementation of the objective function we have settled for the following objective function types:

Quadratic As is common in optimal control, use a functional in which the squared difference between a variable and its corresponding set point:

$$Q_{\mathtt{sp}}(t) = \int_{t=0}^{T} w_{\mathtt{sp}}(y(t) - y_{\mathtt{sp}}(t))^2 dt \tag{4}$$

where $w_{\mathtt{sp}}$ is a weight factor, $y_{\mathtt{sp}}(t) = (\hat{y}(0) - y^0_{\mathtt{sp}})(1 - e^{-t/T_c}) + y^0_{\mathtt{sp}}$ (T_c is the desired response time) and $y^0_{\mathtt{sp}}$ is the desired set point values to which $y(t)$ should approach. This formulation will hopefully, if T_c is chosen carefully, result in a control action which is not to aggressive.

Linear There is also a possibility to have a simple linear objective function:

$$Q_{\mathtt{lin}} = \int_{t=0}^{T} w_{\mathtt{lin}} y(t) dt \tag{5}$$

where $w_{\mathtt{lin}}$ is another weight factor.

Differential A third option is to have a penalty on the squared differential value of the manipulated variables:

$$Q_{\mathtt{d}} = \int_{t=0}^{T} (\dot{u}(t))^2 dt \tag{6}$$

with the weight factor $w_{\mathtt{d}}$. This term is motivated by the wish to reduce large variations in the manipulated variables.

For the first two objective function types, it is possible to replace y with x, z or u, where appropriate. For the optimization formulation, it is also meaningful to have (time-dependent) constraints on all variables, i.e.:

$$x_l(t) \le x(t) \le x_u(t) \tag{7}$$
$$u_l(t) \le u(t) \le u_u(t) \tag{8}$$
$$z_l(t) \le z(t) \le z_u(t) \tag{9}$$
$$y_l(t) \le y(t) \le y_u(t) \tag{10}$$

We also consider hard constraints on the rate of change for the manipulated variables,

$$u^c_l \le \dot{u}(t) \le u^c_l \tag{11}$$

such that impossible changes in the manipulated variables are avoided.

We thus have a continuous-time optimal control problem (CTOCP), defined by the objective functions 4-6 and subject to the constraints 1-3 and 7-11, which we need to solve.

In our solution, we follow the approach of solving the DAE system simultaneously with the optimization criteria 4-6. This is performed by the discretization of the DAE and the objective function and cast them into a large non-linear programming concept. As a discretization method, a second order variable step-size version of the backward differentiation formula (BDF) is used. This results in the following non-linear program:

$$\min_{u,x,y} Q(u,x,z) = Q_{\mathtt{sp}} + Q_{\mathtt{lin}} + Q_{\mathtt{d}} \tag{12}$$

s.t.

$$0 = g(x_k, z_k, u_k, p) \tag{13}$$
$$0 = \alpha_{k2} x_{k-2} + \alpha_{k1} x_{k-1} + \beta_k f(x_k, z_k, u_k, p) \tag{14}$$
$$0 = y_k - h(x_k, z_k, u_k, p) \tag{15}$$
$$0 = u^s_k + u_k - u_{k-1} \tag{16}$$
$$0 = u_0 - \hat{u}_0 \tag{17}$$
$$0 = x_0 - \hat{x}_0 \tag{18}$$
$$\tag{19}$$

and

$$x^l_k \le x_k \le x^u_k \tag{20}$$

$$z_k^l \leq z_k \leq z_k^u \tag{21}$$
$$y_k^l \leq y_k \leq y_k^u \tag{22}$$
$$u_k^l \leq u_k \leq u_k^u \tag{23}$$
$$u_s^l \leq u_k^s \leq u_s^u,\ k = 1, N \tag{24}$$

where

$$Q_{\mathtt{sp}} = \sum_{k=1}^{N} w_{\mathtt{sp}}(y_k - y_{k,\mathtt{sp}})^2 \tag{25}$$
$$Q_{\mathtt{lin}} = \sum_{k=1}^{N} w_{\mathtt{lin}} y_k \tag{26}$$
$$Q_{\mathtt{d}} = \sum_{k=1}^{N} w_{\mathtt{d}}(u_{k-1} - u_k)^2 \tag{27}$$

The parameters α_{k1}, α_{k2} and β_k are the step-dependent BDF weights (Ascher and Petzold 1998).

To solve this NLP it is possible to utilize the time-stage characteristics, see e.g. (Franke and Arnold 1997) and (Cervantes *et al.* 2000). However, we have chosen to lump the optimization related variables into the vector χ, such that

$$\chi \equiv \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ z_1 \\ y_1 \\ u_1^s \\ \vdots \\ x_N \\ u_N \\ z_N \\ y_N \\ u_N^s \end{bmatrix} \tag{28}$$

The NLP can now be defined by

$$\min_{\chi} Q(\chi) \tag{29}$$

s.t.

$$c(\chi) = 0 \tag{30}$$
$$\chi^l \leq \chi \leq \chi^u \tag{31}$$

where $c(\cdot)$ are the Equations 14-19. The constraints are defined with the same ordering as in Equation 28. With this notation, any general solver of NLP can be used.

4. STATE ESTIMATION

In order to perform a meaningful production planning it is necessary to know the current state of the process, i.e. $\hat{x}_0$ and $\hat{u}_0$. Some of the state variables are directly measured, most notably the tank levels and pulp concentration at some positions, while many other are not. It is thus necessary to estimate the state of the process by some statistical method. For linear dynamic systems the classical Kalman filter provides an optimal estimate of the state. Kalman filter based methods have been used to estimate the state also for non-linear systems described by ordinary differential equations (Anderson and Moore 1979) and by differential algebraic equations (Becerra *et al.* 2001). However, these methods requires some manual and analytical manipulation of the original DAE system, which seemed impossible given the mere size of our problem. Instead, a moving horizon estimation (MHE) approach, described in (Rao 2000), was taken. MHE methods also has the advantage over Kalman filter based methods in that it allows for hard constraints on the variables, thus ensuring that the solution is always physically reasonable. It may be mentioned that for real-time applications, where the solution time of the state-estimation is very critical, MHE is probably not suitable, but in our application this is really not a problem.

Given the system of DAE:s from Section 2, we augment it with process and measurement noise:

$$0 = g(x(t), z(t), u(t), p) \tag{32}$$
$$\dot{x}(t) = f(x(t), z(t), u(t), p) + w(t) \tag{33}$$
$$y(t) = h(x(t), z(t), u(t), p) + v(t) \tag{34}$$

where $w(t)$ and $v(t)$ are vectors of independently uncorrelated random variables with zero mean and covariance matrices Q and R, respectively. It must be mentioned that we are familiar with the fact that Equation 33 is, from a strict mathematical point of view, nonsense. However, our excuse is that in the control community in general and in the chemical engineering community in particular, it is common practice to use this notation.

To estimate the state we discretize (32)-(34) and add upper and lower limits on the variables. The objective is is based on minimizing the squared estimated values of the process and measurement noise. There is also an extra term in the objective to penalize the deviation of the estimated initial state from the prior estimated state. We thus have a NLP formulated as follows:

$$\min[||x_{T-N} - \bar{x}_{T-N}||^2_{\Pi^{-1}} + \sum_{k=T-N}^{T} (||w_k||^2_{Q^{-1}} + ||v_k||^2_{R^{-1}})] \tag{35}$$

s.t.

$$0 = g(x_k, z_k, u_k, p) \tag{36}$$

$$0 = \alpha_{k2}x_{k-2} + \alpha_{k1}x_{k-1} + \beta_k f(x_k, z_k, u_k, p) + w_k \quad (37)$$

$$0 = y_k - h(x_k, z_k, u_k, p) + v_k \quad (38)$$

$$0 = y_k^m - y_k,\ k = T-N, T \quad (39)$$

where Q, R and Π are diagonal matrices of appropriate dimensions, respectively. The vector y_k^m contains the measurements at the interval k. The size of the estimation window, $[0, N]$ is usually determined by a a trade-off between the required computational effort and the accuracy, respectively. If a measurement value is missing, or invalid in some other way at an time interval, the corresponding equation in Equation 39 is removed. Notice also that we are using hard constraints on the measurement and process noise, respectively. The vector $\bar{x}_{T-N}$ is the estimated state from the previous estimation.

From the NLP (35)-(39), we see that except for the objective function and the additive noise, the formulation of the constraints is similar to the production planning problem, thus enabling the re-use of much of the code implementation. Also for this problem, we lump the free variables in a single vector:

$$\chi_e \equiv \begin{bmatrix} x_{T-N} \\ x_{T-N+1} \\ u_{T-N+1} \\ z_{T-N+1} \\ y_{T-N+1} \\ v_{T-N+1} \\ w_{T-N+1} \\ \vdots \\ x_T \\ u_T \\ z_T \\ y_T \\ v_T \\ w_T \\ p \end{bmatrix} \quad (40)$$

and make the general formulation:

$$\min_{\chi_e} Q_e(\chi_e) \quad (41)$$

s.t.

$$c_e(\chi_e) = 0 \quad (42)$$

$$\chi_e^l \le \chi_e \le \chi_e^u \quad (43)$$

where the annoying subscript e refers to the estimation problem.

For an on-line application the MHE works as follows:

(1) At start-up, the initial values $\bar{x}_{T-N}$ are chosen to some standard or nominal values.
(2) The estimation window is calculated from the estimation horizon and the current time T.
(3) Measured values are collected for the estimation window and re-sampled to represent the N discrete time intervals.
(4) Solve the MHE problem (35)-(39).
(5) Set $T := T + 1$ and goto 2.

5. SOLUTION OF THE NON-LINEAR PROGRAMS

The previous sections formulated two large non-linear programs as the solution of the production control and state-estimation problems, respectively. To solve these NLP, two different solvers for large scale optimization problems have been used. The first one, SNOPT, is a software based on a sequential quadratic programming (SQP) method (Gill *et al.* 2000). SNOPT uses a limited memory quasi-Newton method to approximate the Hessian of the Lagrangian and uses a line search based on an augmented Lagrangian merit function. The user provides SNOPT with objective and constraint functions and their gradients. It also allows for the user to explicitly specify which variables and equations that enters the constraints and objective linearly. Thus, SNOPT is claimed to be more effective when most constraints and variables are linear.

The second solver, IPOPT, is based on a primal-dual algorithm interior point algorithm (Wachter and Biegler 2004). Here the user provides the objective function, constraint function, their gradients and the analytical Hessian of the Lagrangian. A filter line search is used to find proper descent directions.

6. PRACTICAL EXPERIENCE WITH THE PRODUCTION PLANNING

This section will make a brief description on the on-line application is performing at Billerud Gruvön. All relevant information is stored in a SQL data base. For example, measured values from the process sensors are collected using an OPC connection to the mills information system and stored as time series in the data base. All instances of the process object models are stored in the database. Composite models are also stored in the data base with information on which objects that are part of the model and how they are connected.

This vast information is accessed through a graphical user interface (UI). Here the user may change values on the instanced objects, create new objects, change the connection configuration, etc.

To perform a state-estimation or an production control optimization, there are two options. Either the user performs an off-line calculation where he can choose the starting time arbitrarily, or he can configure a scheduled task list, which is performed on a chosen regular basis. The task list defines a series of calculations to be performed, typically it involves a state-estimation to provide the production control optimization which a consistent initial value. The frequency of schedule must be chosen such that the calculation if finished before the next job is to start, otherwise the previous job is terminated and a new one is started.

Although the scheduled solution of the production control optimization is the main deliverable of this project, the off-line calculation possibilities proved to be very valuable during the engineering phase of the project, e.g. for fitting parameters in the models and for tuning of the production control strategy.

During the period from May 2004, when the final version of the production planning system was installed, until September 2004, the system has been in scheduled action. The scheduler is set on a cycle in which a state estimation with a horizon of 5 hours is solved first, followed by a production optimization with a horizon of 40 hours. This corresponds to NLP:s with 15000 and 30000 variables, respectively. The cycle is repeated every 40 minutes. The average computational times are 30 seconds for the state estimation and 15 minutes for the production optimization. The rate of successful solutions is 95% for the state estimation and 90% for the production optimization, respectively. The main reason for not being able to solve the problem is to the computational time, which occationally is dramatically increased, compared to the average values.

7. SUMMARY AND CONCLUSIONS

This paper has presented results from the development and implementation of methods for the solution of optimal control problems applied to systems described by differential algebraic equations. A framework for the specification of models and the subsequent generation of Modelica libraries and solver specific Fortran code. A pilot installation has been done at Billerud Gruvön.

Two state-of the art solvers, SNOPT and IPOPT, are integrated for the solution of the resulting large-scale non-linear programs. Pre-processing is done in order to find a suitable start vector for the optimization, resulting in an acceptable and robust performance. The production control tool has been running on-line at Gruvön for several months, scheduled to solve a state-estimation and a production planning problem every 40 minutes.

The main conclusion that so far can be made is that it is indeed possible to on-line solve the production planning problem using optimization methods. The proposed production trajectories are generally accepted by the production engineers, although it is to early to summarize the actual benefit for the mill. More quantitative analysis and results on the system performance will most surely be available during the next 6 months of operation.

REFERENCES

Anderson, B.D.O. and J.B. Moore (1979). *Optimal Filering*. Prentice Hall.

Ascher, U. and L. Petzold (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM Press.

Becerra, V. M., P. D. Roberts and G. W. Griffiths (2001). Applying the extended kalman filter to systems described by nonlinear differential-algebraic equations. *Chem. Eng. Science* pp. 267–281.

Cervantes, A., S. Tonelli, A. Brandolin, A. Bandoni and L. Biegler (2000). Large-scale dynamic optimization of a low density polyethylene plant. *Comp. Chem. Engineering* **24**, 983–989.

Franke, R. and E. Arnold (1997). Applying new numerical algorithms to the solution of discrete-time optimal control problems. In: *Computer-Intensive Methods in Control and Signal Processing: The Curse of Dimensionality* (K. Warwick and M. Karny, Eds.). pp. 105–118. Birkhauser. Basel.

Gill, P., W. Murrey and M. Saunders (2000). Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM J. on Optimization* **12**, 979–1006.

Rao, C.V. (2000). Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems. PhD thesis. University of Wisconsin.

Wachter, A. and L.T. Biegler (2004). On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Technical report. IBM T. J. Watson Research Center.