A Mixed Integer Moving Horizon Formulation for Prioritized Objective Inferential Control of a Bioprocess System

C. E. Long†, E. O. Voit‡, and E. P. Gatzke†[1]
†Dept. of Chemical Engineering, University of South Carolina, Columbia, SC 29208
‡Dept. of Biometry and Epidemiology, Medical University of South Carolina, Charleston, SC 29425

Abstract

This paper presents a Model Predictive Control (MPC) algorithm utilizing a state-space approach that allows for inferential control of unmeasured states using a prioritized control objective formulation. Knowledge of the unmeasured states is gained through the use of an external state estimation routine, while mixed-integer methods are used to implement the prioritization of the objectives. The capabilities of the algorithm are demonstrated by the application of the controller to a fermentation reactor model in a simulation environment.

1 Introduction

A model of a dynamic system may have many states that describe the dynamic response of the system. However, at any given time some of these states may not be known. Some process states may be too costly, too time consuming, or simply impossible to directly measure. This lack of available online measurements is obvious in the case of complex metabolic and genomic systems, where many of the molecular species and reactions are only accessible with significant experimental effort. Nevertheless, it is often desirable to control these unmeasured states in a systematic manner, especially during reference change transitions.

Model Predictive Control (MPC) methods have become quite popular in industry because of their ability to control a broad range of processes. For a detailed review of MPC, see references [1, 2, 3]. These methods are capable of handling multivariable systems, enforcing hard and soft constraints on both inputs and outputs, and accommodating process time delays and difficult dynamics. The controller operates by using process data at discrete time intervals to formulate an optimization problem, which represents the minimization of some cost (objective) function. The optimization problem is solved at a given time step for the optimal control trajectory and the appropriate control move is implemented. This is repeated at each time step to provide online real time control. The MPC works well under the premise that three goals can be met. First, the system in question must be modeled in an effective manner. Second, the proper optimization problem must be formulated using data from that model. Finally, the optimization problem must be solved efficiently to yield the necessary control move in the allotted sampling time. Variations in these three steps affect the efficiency of a particular MPC method.

This paper demonstrates that an MPC can be designed to accommodate the needs of a system that has unmeasured states that need to be regulated. Such a design benefits from a state-space formulation with prioritized control objectives and leads to a standard mixed-integer quadratic programming problem. Specifically, a conventional MPC algorithm minimizes an objective function with weights for penalizing both setpoint errors and rapid changes in input level. The problem is typically constrained according to needs from the problem's domain. This often requires setpoint constraints on process outputs, one-sided constraints for soft limits on process variable upper and lower bounds, and constraints on the input movements for actuator limitations. In contrast to a conventional MPC, the use of a state-space model along with an external state estimation routine can explicitly define all states for the process across the entire time horizon. This way, any unmeasured model states that may enter into unsafe or undesirable operating regions can be inferentially manipulated. This is accomplished by incorporating into the optimization problem constraints for unmeasured state estimates in the MPC prediction horizon projection, just as one traditionally deals with process outputs. In the traditional MPC method, all variables are continuous and the optimization problem is formulated either as a linear (LP) or a quadratic program (QP) depending on the norm used to characterize deviations from the ideal operating point of the system.

Mixed integer formulations have been presented in the past both to discretize and prioritize the traditionally continuous control objectives [4, 5, 6]. One advantage of methods based on propositional logic is that control objectives are explicitly stated and prioritized, avoiding some uncertainties associated with MPC controller tuning. The main disadvantage of these methods is the computational complexity of the op-

[1]Corresponding author's email: gatzke@sc.edu

timization problem to be solved online. In mixed-integer formulations, the objective function is modified and additional constraints are added that describe whether discrete objectives have been met at all and whether they were met in a pre-specified ranking of priority. Large objective function penalties are assigned if the discretized objectives are not being met, and even larger penalties are placed on not meeting objectives in order of their priority. These penalties are added to those used in the conventional MPC formulation. Using the state-space formulation offers the additional advantage that constraints can be placed on the explicitly defined unmeasured states and errors associated with these states can be penalized in the objective function, just like they are for measured states. Formulation of the objective function with priority control incorporates both binary and continuous variables, as opposed to exclusive use of continuous variables in the conventional case. The result of this formulation is a standard mixed-integer optimization problem that is either linear (MILP) or quadratic (MIQP).

A time-efficient solution of these optimization problems is crucial for real time control, but this issue is beyond the scope of this article. Given that the MPC task consists of solving a new optimization problem at each time step, the limitation of the approach becomes whether the posed problem can be solved within the allotted time. Efficient solvers for the LP, QP, MILP, and MIQP problems are available, but the specifics of the problem (such as the total number of variables, the number of binary variables, etc...) dictate whether these solvers are sufficiently efficient for a given application. For the work presented here, the optimization problem is formulated completely in MATLAB [7] and solved using solvers from the IBM OSL Library [8].

This paper presents an MPC algorithm that is capable of controlling unmeasured states and uses a state-space formulation with a prioritized control objective. The specific implementation furthermore allows any subset of the continuous objectives to be discretized. Finally, using the "and" clause from propositional logic [9], multiple discrete objectives can be prioritized in a manner where any number of objectives are assigned the same priority.

The proposed algorithm's capabilities have been tested for a variety of applications in a simulation environment. As an illustration here, simulation results for the operation of a fermentation reactor will be presented. Metabolic reactions inside the cell are modeled using a power-law term for each individual reaction, a strategy which is often referred to as a Generalized Mass Action (GMA) model approach [10]. The fermentation reactor is based on metabolic pathway model with five states, which is represented in GMA format. We suppose that the task consists of controlling a single metabolite concentration (ATP) by manipulating the external glucose concentration in the medium. Various discrete control objectives are used to inferentially constrain concentrations of other metabolites of interest.

2 Controller Formulation

A given process has, or could be simplified to, n_u inputs, n_x states, n_y outputs, and n_d measured disturbances. Consider a linear model of the process in the typical state-space form of:

$$x(k+1) = Ax(k) + B_u u(k) + B_d d(k) \quad (1)$$

$$y(k) = Cx(k) + D_u u(k) + D_d d(k) \quad (2)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state vector at sample time k, $u \in \mathbb{R}^{n_u}$ is the vector of inputs, $y \in \mathbb{R}^{n_y}$ is a vector of the predicted outputs, and $d \in \mathbb{R}^{n_d}$ includes any measured disturbances.

Using a prioritized objective control approach [4, 6], take the objective function to be minimized at every time step k as:

$$\Phi(k) = \Gamma_p^T P(k) + \Gamma_o^T O(k) + \sum_{i=1}^{p} \left\| \Gamma_e^T e(k+i) \right\| + \sum_{i=0}^{m-1} \left\| \Gamma_u^T \Delta u(k+i) \right\| \quad (3)$$

Here, O and P are vectors of binary elements (0 or 1) that describe whether or not the discretized objectives have been met and if they were met in their specified order (according to their associated priority.) The variables m and p are the controller's move and prediction horizons, e is the vector of errors in either the states or outputs that show the difference between the modeled value and the reference, and Δu is a vector defining the input movements (i.e., the difference between the input positions at two consecutive time steps). Γ_p and Γ_o are vectors of weights corresponding to each discretized objective and to each priority respectively, while Γ_e and $\Gamma_{\Delta u}$ are vectors of weights with entries corresponding to each error term and input move term. Each of these weighting factors provides the ability to assign some relative importance to each individual term within the control problem. The optimization problem to be solved at every time step is constrained. Different types of constraints that could apply are detailed below.

Using the state space approach, all states and outputs are explicitly represented by the model using equality constraints. For states, the constraints are:

$$x(k+i) = Ax(k+i-1) + B_u u(k+i-1) + B_d d(k) \quad (4)$$
$$\forall i = 1 \ldots p$$

For the outputs, the constraints are:

$$y(k+i) = Cx(k+i) + Du(k+i) \quad (5)$$
$$\forall i = 1 \ldots p$$

Note that the controller can only choose input moves over the horizon m, for every i greater than $m-1$, such that the input u has the same value: $u(k=m-1) = u(k+m) = \ldots = u(p)$.

A process can have numerous objectives that are applied to a system using setpoint error constraints. These constraints vary in form depending on the type of constraint required. Traditionally, only the measured states or outputs are constrained, but as mentioned before, it could be important for the controller to control some of the unmeasured states. To this end, the controller formulation must have the ability to constrain both types of states (unmeasured and measured), as well as the outputs. This can be achieved either with a setpoint constraint or a one-sided soft upper or lower bound. A setpoint constraint on a state is written as:

$$\left|r_{x_j}(i) - x_j(i)\right| \leq e_x(i) + B_j \quad \forall i = 1...p \tag{6}$$

where r_{x_j} is the reference value from a reference vector r for the j^{th} state and B_j is a tolerance value that provides a range within which the state must stay in order for the constraint to be satisfied. This single constraint can be split up into two separate constraints

$$\begin{array}{rcrcrcl} & r_{x_j}(i) & - & x_j(i) & \leq & e_x(i) & + & B_j \\ - & r_{x_j}(i) & + & x_j(i) & \leq & e_x(i) & + & B_j \end{array} \qquad \forall i = 1...p \tag{7}$$

The soft upper or lower bound can be placed on a state by enforcing only one side of the setpoint constraint defined in Equation 7. The objective function contains a term that penalizes input movements. These input movements are measured and constrained by

$$|u(k+i-1) - u(k+i-2)| \leq \Delta u(k+i-1) \quad \forall i = 1...m \tag{8}$$

This is the last constraint for the traditional MPC algorithms. The following constraints are imposed in addition and pertain specifically to the prioritized objective formulation.

Discretizing the typical continuous control objectives immediately renders it possible to prioritize different objectives through a series of algebraic constraints. To discretize the continuous control objectives, a constraint of the form:

$$e(i) \leq N_j(1 - O_j) \quad \forall i = 1...p \tag{9}$$

is needed where N_j is a very large value. If the continuous control objective has a non-positive error at all time steps over the prediction horizon, the objective is satisfied and O_j will take on a value of one. On the other hand, if the control objective is not satisfied at one of the time steps over the prediction horizon (i.e., there is a positive error at some time), the O_j will take on a value of zero for the constraint to be satisfied.

To ensure that the objectives are met in order of priority, another constraint is introduced. Previous work [4] in this area used a formulation in which all continuous objectives were discretized and each objective was given a different priority. In this case, the constraints were of the form: $P_i \leq O_i$ for all $i = 1...Np$, where N_p and N_O are the number of priorities and objectives respectively and $N_p = N_O$. Using the "and" clause from propositional logic [9], it is quite easy to extend this type of constraint to cases where several objectives have equal priority ($N_p \neq N_O$). Specifically , one constraint is added for each objective that forces the objectives to be met before the corresponding priority. As an example, one might require:

$$\begin{array}{rcl} P_1 & \leq & O_1 \\ P_1 & \leq & O_2 \\ P_2 & \leq & O_3 \\ \vdots & & \vdots \\ P_{N_p} & \leq & O_{N_O} \end{array} \tag{10}$$

In this case, the first two objectives have the same priority, therefore both O_1 and O_2 must be met before P_1 is satisfied. An extension to the case of equal priorities is simply a matter of propositional logic. Similar extensions can furthermore handle any logical clause consisting of combinations of "and" and "or" statements.

Finally, constraints are needed to enforce the order in which the objectives are met based on their priorities. These constraints are:

$$P_{i+1} \leq P_i \quad \forall i = 1...N_p - 1 \tag{11}$$

All of the elements of the problem are now defined, but it is still necessary to express them in a more general and compact form that is compatible with the solver. The objective function can be rewritten in the form:

$$J = \frac{1}{2} z^T H z + f^T z \tag{12}$$

H is a diagonal weight matrix that holds the weights of each term whose magnitude is measured by the l_2-norm and f is a vector holding the weights of each term whose magnitude is measured by either the l_1-norm or l_∞-norm. When H is nonzero, the optimization problem is a constrained Mixed Integer Quadratic Programming (MIQP) problem. However, if $H = 0$, the problem is simplified to a constrained MILP problem. In the objective function, z is a vector of the unknowns. It is defined as:

$$z = [u_m^T\ x_p^T\ y_p^T\ e_p^T\ \Delta u_m^T\ O^T\ P^T]^T \tag{13}$$

where:

$$u_m = [u(0)^T ... u(m-1)^T]^T$$

$$x_p = [x(1)^T ... x(p)^T]^T$$

$$y_p = [y(1)^T ... y(p)^T]^T$$

$$e_p = [e(1)^T ... e(p)^T]^T$$

$$\Delta u_m = [\Delta u(0)^T ... \Delta u(m-1)^T]^T$$

It should be noted that from this unknown vector, the controller only has $m * n_u$ true decision variables. Only u_m must be specified. With the value of u_m and the current process measurements, the rest of z is defined by the constraint relations (Eqs. 4-11).

All constraints can then be rearranged into a single mathematical expression in the standard matrix-vector notation of:

$$Mz \leq b \tag{14}$$

where, again, z is the vector of unknowns, M is a matrix that holds the constant constraint parts, and b is the vector that holds the portions updated by process measurements. The optimization problem posed and solved by the controller is then:

$$min \frac{1}{2} z^T H z + f^T z \tag{15}$$

subject to $Mz \leq b$ and $z_{lb} \leq z \leq z_{ub}$. The selection of lower and upper bounds on z (z_{lb} and z_{ub}) allow for hard constraints, such as actuator limits, to be imposed on the system in addition to those constraints detailed before. The desired control action is taken, and then the controller waits for the next sample time to receive the updated process data. These updated data are the basis for formulating the appropriate optimization problem for the next time step.

3 Application to a Fermentation Reactor

3.1 The System

The anaerobic fermentation pathway of yeast *(Saccharomyces cerevisiae)* has been the subject of numerous studies over the years. In this pathway, yeast takes up glucose from the medium and, after a series of intermediate reactions, produces ethanol and a number of other metabolites. A simplified schematic of the pathway, as proposed by Curto et al (1995), is shown in Figure 1.

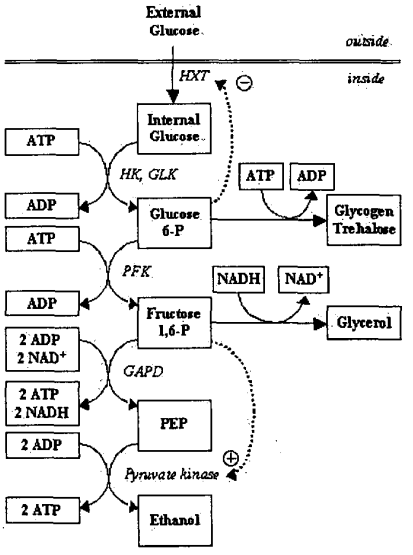

Figure 1: Simplified model of anaerobic fermentation of glucose to ethanol, glycerol and polysaccharides in Saccharomyces cerevisiae. Solid arrows represent reactions and dotted arrows show modulations. State variables in the model are: X_1 = cytosolic glucose; X_2 = glucose-6-phosphate; X_3 = fructose-1,6-diphosphate; X_4 = phosphoenol pyruvate; X_5 = ATP. Independent variables with constant values are: X_6 = hexose transport; X_7 = hexokinase/glucokinase; X_8 = phosphofructokinase; X_9 = glyceraldehyde dehydrogenase; X_{10} = pyruvate kinase; X_{11} = glycogen and trehalose production; X_{12} = glycerol production; X_{13} = ATPase; X_{14} = NADH/NAD+ ratio.

The metabolic reactions inside the cell can be modeled in a variety of ways. One strategy consists of presenting collectively all reaction rates of species entering a pool or leaving a pool by products of power-law functions that contain all contributing metabolites, enzymes, and modifiers. Thus, each differential equation in the coupled set consists of a single difference between two products of power-law functions. This type of model is often referred to as an S-system model and falls into the domain of Biochemical Systems Theory [10, 11, 12, 13, 14]. Alternatively, each individual metabolic reaction can be modeled with its own power-law, which yields a so-called Generalized Mass Action (GMA) model and is closer to biochemical intuition than the S-system representation. For this work, a fermentation reactor based on a five state GMA metabolic pathway model [10, 11] is considered. The five states are defined by the set of differential equations provided at the top of the next page. The states ($X_1, X_2, X_3, X_4,$ *and* X_5) of the system are five of the intermediate metabolite concentrations internal to the cell as shown in Figure 1. For illustration purposes, it is assumed that only a single measurement is available, namely the concentration of ATP in the system. The single output is $y = X_5$.

$$
\begin{aligned}
\dot{X}_1 &= 0.8122X_2^{-0.2344}X_6 - 2.8632X_1^{0.7464}X_5^{0.0243}X_7 \\
\dot{X}_2 &= 2.8632X_1^{0.7464}X_5^{0.0243}X_7 \\
&\quad -0.5232X_2^{0.7318}X_5^{-0.3941}X_8 - 0.0009X_2^{8.6107}X_{11} \\
\dot{X}_3 &= 0.5232X_2^{0.7318}X_5^{-0.3941}X_8 \\
&\quad -0.011X_3^{0.6159}X_5^{0.1308}X_9X_{14}^{-0.6088} \\
&\quad -0.04725X_3^{0.05}X_4^{0.533}X_5^{-0.0822}X_{12} \\
\dot{X}_4 &= 0.022X_3^{0.6159}X_5^{0.1308}X_9X_{14}^{-0.6088} \\
&\quad -0.0945X_3^{0.05}X_4^{0.533}X_5^{-0.0822}X_{10} \\
\dot{X}_5 &= 0.022X_3^{0.6159}X_5^{0.1308}X_9X_{14}^{-0.6088} \\
&\quad +0.0945X_3^{0.05}X_4^{0.533}X_5^{-0.0822}X_{10} \\
&\quad -2.8632X_1^{0.7464}X_5^{0.0243}X_7 - 0.0009X_2^{8.6107}X_{11} \\
&\quad -0.5232X_2^{0.7318}X_5^{-0.3941}X_8 - X_5X_{13}
\end{aligned}
$$

This leaves the remaining four states unmeasured. The only input (u) to the system is the rate of glucose uptake (X_6) by the cell. Two possible measured disturbances are taken as the rate of polysaccharide production (glycogen and trehalose) and the rate of glycerol production (X_{12}). All remaining variablesin the GMA model are held constant in order to simplify the problem for demonstration purposes. Further details of the system can be found in [10, 11]. This GMA model will act as the process to be controlled by the MPC.

3.2 Controller Tunings and Specifics

The controller relies on a model of the system both to estimate and constrain the states and outputs. A continuous-time linear state-space model of the system was obtained using a perturbation-based algorithm in the simulation environment. This algorithm linearizes the system around a single steady-state operating point. The steady-state operating point was chosen so that $u_{ss} = 19.7$ mM/min and that the two metabolic production rates, which are used as disturbances of the system at steady state, had values of 14.31 and 203 mM/min respectively. The state vector was found to have a steady state value of:

$$x_{ss} = [0.3458,\ 1.0099,\ 9.1969,\ 0.00953,\ 1.1247]^T$$

The continuous time model was then discretized using a zero-order hold approach and a sampling rate of 0.1 min. From the resulting model it was determined that the five-state system is completely observable with the single measurement of ATP concentration mentioned above. A Luenberger observer was created and used as a means to estimate the states. The observer poles were placed far enough inside the unit disk to provide sufficiently fast convergence of the state estimates to the actual state values.

It should be noted that numerous states of this system can exhibit an inverse response. In order for the MPC to accomodate this dynamic system, the prediction horizon (p) must be chosen large enough so that the controller can view the state estimates far enough into the future. For this illustration we chose $m = 2$ and $p = 30$. The weights on all continuous error variables that arise from the traditional setpoint and from the soft upper and lower bounds are set so that the diagonal elements of $\Gamma_e = 100$. The weights on the size of a single input movement are defined by $\Gamma_{\Delta u} = 50$. The elements of Γ_o and Γ_p are set to values of -1000 and -2000 respectively.

3.3 Controller Performance

To demonstrate the closed-loop performance of the controller, both its ability to handle a reference transition and a disturbance load are shown. Specifically, a setpoint transition is made on the output of the system (ATP concentration) where the output is stepped from its steady state value of $1.1247\,mM$ to $1.2\,mM$ at $t = 5min$. A setpoint constraint is utilized to move this concentration with its setpoint. At $t = 25min$, a disturbance load is imposed by stepping the rate of glycerol production from its normal operating point of $203mM/min$ down to $50mM/min$. All the while, a lower bound is placed on the output $0.05\,mM$ below the setpoint to prevent the inverse response of the system from taking this output to a undesirably low value. An upper bound is placed $0.02mM$ above the new setpoint. The third state of the system, the unmeasured concentration of fructose-1,6-phosphate, is also constrained by both a soft upper and a lower bound. The upper bound is placed at a level of $10\,mM$ and the lower bound is placed at a concentration of $9\,mM$. It should be noted that enforcement of all constraints is delayed by one time step in all cases. Details of the constraints are provided in Table 1. For this particular problem, there are 490 constraints and 341 variables, 7 of which are binary.

Table 1: Summary of Discrete Objectives for the Fermentation Pathway System

Constraint Type	Constraint	Disc. Obj. Number	Priority Number
Upper Bnd	$-r_{x_3} + x_3 \leq 0.7131$	1	1
Upper Bnd	$-r_y + y \leq 0.05$	2	1
Lower Bnd	$r_y - y \leq 0.05$	3	2
Lower Bnd	$r_{x_3} - x_3 \leq 0.1969$	4	3

The closed-loop performance of the system is presented in Figure 2. When the reference transition occurs, the nature of the system response wants to take the output in the opposite direction, and at this time, not all of the control objectives can be met: the third discrete objective (lower bound on the output) is violated. The controller actually rushes to get this value back above this constraint and does so in just a few time steps. If this constraint were not present, the inverse response could have taken the output to an even lower value for a substantial amount of time. This objective is of priority two, but even though it could not be met, the controller successfully attempted to satisfy the fourth objective

which is of lower priority. Figure 3 shows that three of four objectives have been met but only those of the highest priority were met in order. After the transient associated with the reference transition, the system has not been able to realize the desired new setpoint. The upper bound constraint placed on the unmeasured state is active and effectively limits the level to which the output can reach at these operating conditions. All of the discretized objectives are met here.

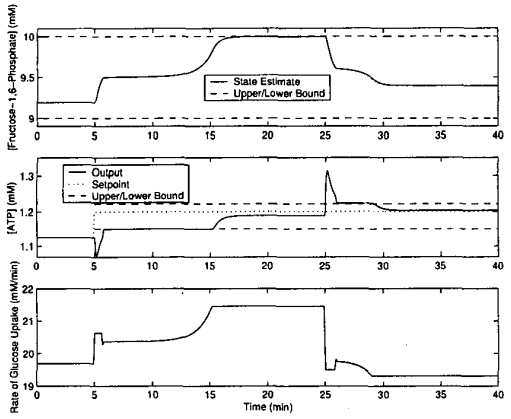

Figure 2: State, Output, and Input Trajectories for the Closed-Loop System after a Reference Transition and a Disturbance Load

At $t = 25min$, the disturbance in carbohydrate production is imposed. This disturbance is so large that the output is pushed above its upper bound for a brief moment. This upper bound is of the highest priority. Note, however, that the first objective, which is of the same priority, is met. Here, the remaining objectives can be met, but not in order of priority, so none of the priority variables are satisfied (Figure 3). The controller aggressively moves the output back down below this level, and all discrete objectives are met. The controller then focuses solely on satisfying the traditional (non-discretized) setpoint constraint.

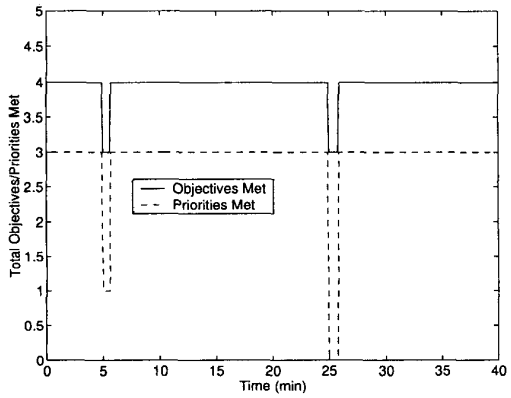

Figure 3: Plot of the Total Number of Discrete Objectives Met and the Number Met in Order of Their Priority

In summary, an MPC algorithm has been presented that utilizes a linear state-space model and explicitly defines all states of a system, including those that are typically unmeasured. Knowledge of the unmeasured states allows these to be inferentially controlled using a flexible mixed integer prioritized objective formulation. The formulation provides the ability to discretize any or all of the constraints and also permits the same priority level for any number of discrete objectives. This design was shown to be effective in the case of a fermentation pathway system. The ATP concentration was moved to its desired setpoint, while the unmeasured state was maintained within some specified bounds.

References

[1] C. E. García, D. M. Prett, and M. Morari. Model Predictive Control: Theory and Practice - A Survey. 25(3):335–348, 1989.

[2] M. Morari and J. H. Lee. Model Predictive Control: The Good, the Bad, and the Ugly. pages 419–444, Padre Island, TX, 1991.

[3] M. Morari and J. H. Lee. Model Predictive Control: Past, Present and Future. In *PSE ESCAPE-7 Symposium*, Trondheim, Norway, 1997.

[4] E. P. Gatzke and F. J. Doyle III. Model Predictive Control Of A Granulation System Using Soft Output Constraints and Prioritized Control Objectives. *Powder Technology*, 121:149–158, 2001.

[5] R. S. Parker, E. P. Gatzke, and F. J. Doyle III. Advanced Model Predictive Control (MPC) for Type I Diabetic Patient Blood Glucose Control. Chicago, IL, 2000.

[6] A. Bemporad and M. Morari. Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica*, 35:407–427, 1999.

[7] The MathWorks. *Matlab 6.1*. Prentice Hall, 2000.

[8] I. B. M. IBM Optimization Solutions and Library Linear Programming Solutions. Technical report, I. B. M., 1997.

[9] M. L. Tyler and M. Morari. Propositional Logic in Control and Monitoring Problems. *Automatica*, 35:565–582, 1999.

[10] E. O. Voit. *Computational Analysis of Biochemical Systems*. Cambridge University Press, New York, 2000.

[11] R. Curto, A. Sorribas, and M. Cascante. Comparative Characterization of the Fermentation Pathway of Saccharomyces cerevisiae using Biochemical Systems Theory and Metabolic Control Analysis: Model Definition and Nomenclature. *Math. Biosc.*, (130):25–50, 1995.

[12] M. A. Savageau. Biochemical Systems Analysis, I. Some Mathematical Properties of the Rate Law for the Component Enzymatic Reactions. *J. Theor. Biol.*, 25(365-369), 1969.

[13] M. A. Savageau. Biochemical Systems Analysis, II. The Steady-state Solutions for an n-pool System using a Power-Law Approximation. *J. Theor. Biol.*, 25:370–379, 1969.

[14] M. A. Savageau. *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. Addison-Wesley, Massachusetts, 1976.