ADCHEM 2006

International Symposium on Advanced Control of Chemical Processes
Gramado, Brazil – April 2-5, 2006

REAL-TIME DYNAMIC OPTIMIZATION OF NON-LINEAR BATCH SYSTEMS

Nathaniel Peters [*] **Martin Guay** [*] **Darryl DeHaan** [*]

[*] *Queen's University, Kingston, Ontario, Canada*

Abstract: In this paper a methodology for designing and implementing a real time optimizing controller for nonlinear batch processes is discussed. The controller is used to optimize the system input and state trajectories according to a cost function. An interior point method with penalty function is used to incorporate constraints into a modified cost functional, and a Lyapunov based extremum seeking approach is used to compute the trajectory parameters. This technique was previously proposed for optimizing differentially flat systems in a cascade implementation. Smooth trajectories were generated with reduced computing time compared to many optimizations in literature. In this paper the theory is extended to general non-flat nonlinear systems in a true on-line optimization.

Keywords: real-time optimization, batch optimization, on-line optimization

1. INTRODUCTION

Industrial chemical processes can be divided into two categories of production: continuous and fed-batch. Continuous processes are designed to run at steady-state. Examples include oil refining, gas processing, and many chemical processes. To maximize the efficiency and profits from these processes, it is necessary to keep the plant in the operating range under disturbances. The optimization task required to operate such processes is usually performed to achieve disturbance rejection, designing con- trollers to reach and maintain set-point effectively, and keeping the down time to a minimum. Since the operating range is generally very narrow, the system dynamics can often be approximated by linear dynamics. Batch processing, however, provides some very unique challenges. Batch processes have a finite operating time, rather than a continuous operation. The control objective in batch processing is not to reach steady state, but to reach some desired objective by the end of the batch. This usually involves movement through a very wide operating range, and nonlinearities in the system can be very strong. Batch optimization focuses on maximizing the performance objective by finding the corresponding system trajectories. Since batch production is usually of low volume, high value production, optimization of its operation is critical to make the process viable. Examples include pharmaceuticals, specialty chemicals, biological processes, and food production.

To determine the optimal profiles for a batch process, an off-line optimization is often performed to design the best possible profiles to be followed by the batch process. Typically, this is accomplished using some sort of nonlinear programming method. The computations are usually very extensive and must be performed off-line and then implemented with a feedback controller for tracking. However if there are changes to the plant, such as the initial conditions, plant dynamics, or disturbances then the off-line trajectory may no longer correspond to the optimal. Under these circumstances it is desirable to make on-line adjustments to the desired trajectories to reach the new optimal performance.

A significant amount of research has been done in the area of so called on-line optimization. Numerous methods have been investigated including dynamic programming (Hestenes, 1966), (Bellman, 1957), discretization, (Cuthrell and Biegler, 1987), and parametrization, (Visser *et al.*, 2000); and these have been incorporated with a variety of model structures and optimization routines.

Dynamic or nonlinear programming is often used to attempt on-line optimization. The heavy demand for computing time restricts the frequency of updates, and consequently these methods usually result in very discrete on-line changes (look at (Noda *et al.*, 2000), (Zuo and Wu, 2000), (Arpornwichanop *et al.*, 2005), (Loeblein *et al.*, 1999), (Fournier *et al.*, 1999), and (Zhang and Rohani, 2003)). This has been termed a pseudo on-line optimization. To try and compensate for the amount of computing time required, the profiles may be parameterized to known trajectories that have met with success (see (Visser *et al.*, 2000)). In a series of papers by Palanki and Rahman (Palanki *et al.*, 1993), (Palanki and Rahman, 1994), (Rahman and Palanki, 1996), and (Rahman and Palanki, 1998) a method is introduced that provides a geometric approach to handling batch optimization. They show how to develop feedback laws for end-point optimization problems under a variety of state space variations.

Gattu (Gattu and Zafiriou, 1999) used a steepest descent method to define a parameter update law for the system trajectories and then tracked it via a controller. This is an example of the application of gradient methods to batch optimization and forms the basis for the theory presented here. The gradient is used to determine the parameters over the whole batch and introduced measurement feedback through a low level tracking controller. Control design may be difficult depending on the system, so it is desirable to incorporate measurement feedback into the optimization routine.

While the development of this methodology comes from the extremum seeking literature, the methodology is quite close to that of Non-linear Model Predictive Control (DeHaan and Guay, n.d.), (Fontes, 2001), (Magni and Scattolini, 2004). Work has been done on the use of receding horizon NMPC for batch systems (Nagy and Braatz, 2003), (Helbig *et al.*, 1998) and a Lyapunov based approach was used in (Jadbabaie *et al.*, 2001).

In the next section the theory is outlined for a true on-line optimization for general nonlinear dynamical control systems. This theory has been developed from the earlier cascade optimization done in (Guay and Peters, n.d.) for flat systems. This theory outlines the design of a true on-line controller based on the current conditions, and extended to a general non-flat nonlinear case.

2. ON-LINE OPTIMIZATION

We consider a general class of nonlinear dynamical systems of the form:

$$\dot{x} = f(x, u) \tag{1}$$

where $x \in \mathbf{R^n}$ are the state variables and $u \in \mathbf{R^p}$ is the vector of input variables, $f(x) : \mathbf{R^n} \to \mathbf{R^n}$ is a smooth continuous functions of x. There is a vector $u(t) = [u_1 ... u_p]$ of p input variables.

The optimization centers around finding a system trajectory that minimizes the dynamic optimization problem for some cost function. The cost function is expressed as follows

$$J(x, u) = \int_0^T q(x(t), u(t))dt \tag{2}$$

subject to the dynamics in 1 and the following constraints on the path and end-point variables

$$w(x(t), u(t)) \geq 0 \tag{3}$$

$$x(0) = x_0 \tag{4}$$

$$x(T) = x_f \tag{5}$$

It is assumed that a continuous control, $u(t)$ exists that can steer the states from x_0 to x_f over the batch interval $t \in [0, T]$. Although T can be treated as a time-varying parameter, in the following discussion T is considered to be fixed.

The input trajectories are parameterized

$$u(t) = \left[\, u_1 \, \ldots \, u_p(t) \,\right] \tag{6}$$

where

$$u_i(t) = \sum_{i=1}^{N} \theta_{ij} \Xi_{ij}(t) \tag{7}$$

where Ξ are the basis functions and θ_i for $i = 1, \ldots, N$ and $j = 1, \ldots, p$ are the parameters to be determined. The state space equations can be rewritten in terms of θ and the initial conditions. If the input is defined as a polynomial then

$$u_i = \theta^T \phi(t) \tag{8}$$

where the parameters and basis functions are expressed as follows

$$\theta = \left[\, \theta_1 \, \ldots \, \theta_N \,\right] \tag{9}$$

$$\phi(t) = \left[\, 1 \;\, t \, \ldots \, t^{N-1} \,\right]. \tag{10}$$

The states can be expressed as a function of the input trajectories and the current conditions

$$x^p = \alpha(\theta^T \phi(\tau), x^m(t)) \tag{11}$$

where the superscript m denotes a measured quantity, and the superscript p denotes a predicted quantity. An analytical solution for the states may not exist, so the system of differential equations must be solved to determine x^p.

Having defined the structure of admissible input trajectories, the following assumptions are necessary to construct the optimization problem. The constraint set

$$\Omega_c = \{x \in \mathbf{R^n}, u \in \mathbf{R^p} | w(x,u) \geq 0\} \tag{12}$$

describes a convex subset of $\mathbf{R^n}$. It is assumed that the input trajectories evolve on a compact subset Ω of $\mathbf{R^n}$. The cost functional $J : \Omega \rightarrow \mathbf{R}$ is assumed to be convex and continuously differentiable on Ω_c. The cost function $q(x,u)$ is assumed to be sufficiently smooth.

The dynamic optimization problem is now expressed in terms of the parameters as followa

$$\min_{\theta} J = \int_0^T q(\alpha(\theta^T\phi(t), x_0^m), \theta^T\phi(t))dt \tag{13}$$

$$w(\alpha(\theta^T\phi(t), x_0^m), \theta^T\phi(t)) \geq 0 \tag{14}$$

$$\alpha(\theta^T\phi(0), x_0^m) = x_0^m \tag{15}$$

$$\alpha(\theta^T\phi(T), x_0^m) = x_f^p. \tag{16}$$

As in (Guay and Peters, n.d.) an interior point method with penalty function is used to include the constraint costs. An interior point method incorporating a log barrier function enforces the state and input constraints. The boundary conditions are incorporated through a terminal penalty function. In the remaining equations obvious notation has been omitted.

Let the path cost with the log barrier function be expressed as follows

$$L = q - \sum_{i=1}^{\rho} \mu_i log(w_i + \epsilon) \tag{17}$$

The new cost functional with interior point inclusion and a penalty function is defined as follows

$$J_{ip} = \int_0^T L d\tau + M(\alpha(\theta^T\phi(T), x^m(0)) - x_f)^2 \tag{18}$$

where $\mu_i > 0$, $\epsilon_i > 0$ and $M > 0$ are the tuning parameters of the cost functional, with μ and ϵ being taken as small as possible, and M taken as large as possible. It is assumed that the constraint set described by 14 is convex over a set Ω and is also convex over a set Υ in the parameter space. While the focus of this paper is convex problems, if the constraint set is not convex, an infeasible interior-point method can be used. Further details can be found in (Benson *et al.*, 2000*b*), (Benson *et al.*, 2000*a*) and (Benson *et al.*, 2002).

To make the optimization based on the current conditions, the cost needs to be split up into the elapsed and remaining costs.

$$J_{ip} = \int_0^t L^m d\tau + \int_t^T L^p d\tau + M(x^p(T) - x_f)^2 \tag{19}$$

where the first integral represents the actual cost being calculated from the measured states, and the second integral is the predicted cost remaining using the current parameters. The measured cost can be thought of as another state of the system, z, such that

$$z = \int_0^t L^m d\tau \tag{20}$$

and

$$\dot{z} = L^m \tag{21}$$

This leaves a modified cost

$$J_{ip} = z + \int_t^T L^p dt + M(x^p(T) - x_f))^2 \tag{22}$$

A Lyapunov-based approach is used to solve the optimization problem (Guay and Peters, n.d.). Assuming that the cost functional is convex with respect to θ over Υ, then the first order conditions can be applied such that at the optimal parameter set θ^*

$$\nabla J_{ip}(\theta^*) = 0 \tag{23}$$

The Lyapunov function is defined as the cost functional

$$V = J_{ip} \tag{24}$$

and the time derivative is given by

$$\dot{V} = \frac{\partial J_{ip}}{\partial t} + \frac{\partial J_{ip}}{\partial x}\dot{x} + \frac{\partial J_{ip}}{\partial \theta}\dot{\theta} + \frac{\partial J_{ip}}{\partial z}\dot{z} \tag{25}$$

where

$$\frac{\partial J_{ip}}{\partial t} = \int_t^T \frac{\partial L}{\partial x^p}\frac{\partial x^p}{\partial t} d\tau + M(x^p(T) - x_f)\frac{\partial x^p}{\partial t} - L(x^p, u^p)|_t \tag{26}$$

and

$$\frac{\partial J_{ip}}{\partial x}\dot{x} = \int_{t}^{T} \frac{\partial L}{\partial x^p}\frac{\partial x^p}{\partial x}\dot{x}d\tau$$
$$+ M(x^p(T) - x_f)\frac{\partial x^p}{\partial x}\dot{x} \qquad (27)$$

$$\frac{\partial J_{ip}}{\partial z}\dot{z} = \dot{z} = L(x^m, u^m)|_t \qquad (28)$$

where $\frac{\partial J_{ip}}{\partial \theta} == \nabla_\theta J_{ip}$.

The expression (25) can be simplified. Given the current time and conditions the predicted dynamics are subject to the following differential equation

$$\frac{\partial x^p}{\partial \tau} = f(x^p, u^p) \qquad (29)$$

with the initial conditions $x^p(t) = x$. Perturbing either of the initial conditions will perturb the solution vectors tangent to the original unperturbed solution. This can be represented by the following simplifying expression.

$$\frac{\partial x^p}{\partial t} + \frac{\partial x^p}{\partial x}\dot{x} = 0 \qquad (30)$$

Using this simplification, $\dot{V}$ becomes

$$\dot{V} = \nabla_\theta J_{ip}\dot{\theta} + L^m|_t - L^p|_t \qquad (31)$$

By definition $x^p(t) = x^m(t)$ and ensuring that $u^p(t) = u^m(t)$ then

$$\dot{V} = \nabla_\theta J_{ip}\dot{\theta} \qquad (32)$$

Using a straightforward steepest descent approach for the parameter update law

$$\dot{\theta} = -k\nabla_\theta J_{ip}. \qquad (33)$$

Then the final form of the Lyapunov function is

$$\dot{V} = -k\nabla_\theta J_{ip}^T \nabla_\theta J_{ip}. \qquad (34)$$

The Lyapunov function is strictly decreasing except when the gradient is zero (which occurs at the minima, and at the end of the batch).

To avoid divergence of the update law, a projection algorithm was introduced to ensure the parameters remain in a convex set. The properties of the projection algorithm are discussed in (Krstic *et al.*, 1995) and is given below

$$\dot{\theta} = Proj(\theta, \Upsilon) = \begin{cases} \Upsilon, & \text{if} \|\theta\| < \omega_n \\ & \text{or}(\|\theta\| = \omega_n \text{and} \\ & \nabla P(\theta) \leq 0) \\ \Psi, & \text{otherwise} \end{cases} \qquad (35)$$

where $\Psi = \Upsilon - \Upsilon\frac{\varsigma\nabla P(\theta)\nabla P(\theta)^T}{\|\nabla P(\theta)\|_\varsigma^2}, \Upsilon = -k\nabla_\theta J_{ip}$, $P(\theta) = \theta^T\theta - \omega_m \leq 0$, θ is the vector of parameter estimates and ω_m is chosen such that $\| \theta \| \leq \omega_m$.

In the next section an example is explored via simulation. In this example it is assumed that the model is perfect and that there is full state feedback.

3. BIOREACTOR SIMULATION

3.1 Problem Definition

The optimization problem deals with a batch bioreactor discussed in (Mahadevan *et al.*, 2001). The state space equations modelling the reaction are given as follows

$$\dot{x_1} = \frac{\mu_m x_1 x_2}{K_m + x_2} - \frac{u x_1}{x_3}$$
$$\dot{x_2} = \frac{u(S_f - x_2)}{x_3} - \frac{\mu_m x_1 x_2}{Y_{xs}(K_m + x_2)}$$
$$\dot{x_3} = u$$

where x_1 and x_2 are the concentrations of the biomass and substrate respectively; x_3 is the volume, and u is the feed rate. The optimization scheme is to maximize the amount of biomass formed at the end of the batch. The optimization problem is described as follows

$$\min_\theta J = \int_0^T -\left(\frac{\mu_m x_1 x_2}{K_m + x_2} - \frac{u x_1}{x_3}\right) dt$$

$$u_1 \geq 0$$
$$x_3 \leq 10$$
$$10 - u_1 \geq 0$$

The system parameters are summarized in Table 1.

Table 1. Parameters and Initial Conditions

Parameters	Values
S_f	15g/L
K_m	1.2g/L
Y_{xs}	0.4g/g
μ_m	0.5/h
T	7.8h
Initial Conditions	Values
x_{10}	1g/L
x_{20}	0g/L
x_{30}	2L
Algorithm Parameters	Values
μ_1	1E−20
ϵ	1E−8
k	10

3.2 Parametrization

In (Mahadevan *et al.*, 2001), the input profile was parameterized by subdividing the batch into a number of intervals and representing each interval by a fifth order polynomial. The optimization was modified to include the constraint that the volume must reach its maximum value by the end of the batch. The initial guess of the parameters was taken from the solution of a highly constrained problem. The optimization was solved using nonlinear programming.

In this paper, a simple fifth order polynomial profile was arbitrarily selected for the input, with the initial parameters set to zero. The goal here is to show that the optimization can be performed with reasonable computing time, and no off-line analysis of the feasible profile or initial guesses. The input was defined as follows

$$u(t) = \sum_{i=1}^{6} \theta_i \left(\frac{t}{7.8}\right)^{i-1}$$

3.3 Modified Cost

Having selected the parametrization, the next step is to construct the modified cost. The volume constraint is incorporated as a terminal cost, while the input constraints are implemented with log barrier functions. The modified cost function is described below

$$\min_{\theta} J = \int_0^T -\left(\frac{\mu_m x_1 x_2}{K_m + x_2} - \frac{u x_1}{x_3}\right) - \mu(log(u_1 + \epsilon) + log(10 - u_1 + \epsilon))d\tau + (x_3(T) - 10)^2$$

The gradient was computed as discussed in the theory, the details are not included here. The next section will discuss the tuning and algorithm issues. The algorithm parameters for this simulation are in Table 1.

3.4 Algorithm and Computing Issues

Computing time is always an issue when an ODE solver is needed to determine the prediction and sensitivities. The Fortran package ODESSA was used to calculate the model prediction and the first order sensitivities. MATLAB was used to perform the simulation of the closed-loop system.

The run time for this 7.8 hour simulation was approximately 10 seconds, using a 1.6 GHz Centrino Processor, using cost gradient as the update law.

3.5 Results

The dynamic optimization technique was applied to the nominal case presented above. The resulting simulation of the state and input variables profiles can be found in Figure 1. As in (Mahadevan *et al.*, 2001), the optimal final biomass concentration obtained was 4.8 g/L. The technique performs as expected without the need for complex parameterizations and the requirement for partial flatness.

To verify the effect of initial conditions and parameters, various initial batch conditions were tested. Under various parameter guesses, the overall performance was unchanged. New initial conditions resulted in new optimal profiles. Table 2 shows the initial and final cost for several different initial conditions. The method used here provided results comparable to those used in (Mahadevan *et al.*, 2001). However no initial understanding of the optimal structure was needed. A simpler structure was used with all the initial parameters set to zero. Less analysis was needed before running the batch, and the overall algorithm was simpler. This technique can be applied to a variety of problems, with minimal pre-batch analysis.

Table 2. Cost Summary

x_{10} (g/L)	x_{20}g/L (g/L)	x_{30}L (L)	Final Cost
1	0	2	4.8
2	0.5	4	4.5
0.5	0.7	0.5	0.8
4	0.2	7	4.7

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new on-line optimizing controller for nonlinear dynamical systems. Smooth trajectories were generated on-line with feasible computing time to construct optimal trajectories without the need for off-line analysis. In future work, we plan to study the impact of imperfect state measurements and parametric uncertainties.

REFERENCES

Arpornwichanop, A., P. Kittisupakorn and I.M. Mujtaba (2005). On-line dynamic optimization and control strategy for improving the performance of batch reactors. *Chemical Engineering and Processing* **44**(1), 101–114.

Bellman, R.E. (1957). *Dynamic Programming.* Princeton University Press. Princeton.

Benson, H.Y., D.F. Shanno and R.J. Vanderbei (2000*a*). Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions. Technical Report ORFE-00-06. Princeton University.

Benson, H.Y., D.F. Shanno and R.J. Vanderbei (2000*b*). Interior-point methods for nonconvex nonlinear programming: Jamming and comparative numerical testing. Technical Report ORFE-00-02. Princeton University.

Benson, H.Y., D.F. Shanno and R.J. Vanderbei (2002). Interior-point methods for nonconvex nonlinear programming: Complementarity constraints. Technical Report ORFE-02-02. Princeton University.

Cuthrell, J.E. and L.T. Biegler (1987). On the optimization of differential algebraic process systems. *AIChE Journal* **33**(8), 1257–1270.

DeHaan, D. and M. Guay (n.d.). A real-time framework for model predictive control of continuous-time nonlinear systems. In: *Proc. IEEE Conf. on Decision and Control*. accepted for publication.

Fontes, F. (2001). A general framework to design stabilizing nonlinear model predictive controllers. *Systems and Control Letters* **43**(2), 127–143.

Fournier, F., M.A. Latifi and G. Valentin (1999). Methodology of dynamic optimization and optimal control of batch electrochemical reactors. *Chemical Engineering Science* **54**(13-14), 2707–2714.

Gattu, G. and E. Zafiriou (1999). Methodology for on-line setpoint modification for batch reactor control in the presence of modelling error. *Chemical Engineering Journal* **75**(1), 21–29.

Guay, M. and N. Peters (n.d.). Real-time dynamic optimization of nonlinear systems: A flatness-based approach. In: *Proc. IEEE Conf. on Decision and Control*. accepted for publication.

Helbig, A., O. Abel and W. Marquardt (1998). Model predictive control for on-line optimization of semi-batch reactors. In: *Proceedings of the American Control Conference*. pp. 1695–1699.

Hestenes, M. R. (1966). *The calculus of variations and optimal control theory*. Wiley. New York.

Jadbabaie, A., J. Yu and J. Hauser (2001). Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control* **46**(5), 776–782.

Krstic, M., I. Kanellakopoulos and P. Kokotovic (1995). *Nonlinear and Adaptive Control Design*.

Loeblein, C., J.D. Perkins, B. Srinivasan and D. Bonvin (1999). Economic performance analysis in the design of on-line batch optimization systems. *Journal of Process Control* **9**(1), 61–78.

Magni, L. and R. Scattolini (2004). Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Trans. Automat. Contr.* **49**(6), 900–906.

Mahadevan, R., S.K. Agrawal and F.J. Doyle (2001). Differential flatness based nonlinear predicted control of fed-batch bioreactors. *Control Engineering Practice* **9**, 889–899.

Nagy, Z. and R. Braatz (2003). Robust nonlinear model predictive control of batch processes. *AIChE* **49**(7), 1776–1786.

Noda, M., T. Chida, S. Hasebe and I. Hashimoto (2000). On-line optimization system of pilot scale multi-effect batch distillation system. *Computers and Chemical Engineering* **24**(2), 1577–1583.

Palanki, S. and S.A.K.M Rahman (1994). State feedback synthesis for on-line optimization of batch reactors with a non-affine manipulated input and free terminal time. In: *Proceedings of the American Control Conference*. Vol. 3. pp. 3076–3079.

Palanki, S., C. Kravaris and H.Y. Wang (1993). Synthesis of stated feedback laws for end-point optimization in batch processes. *Chemical Engineering Science* **48**(1), 135–151.

Rahman, A.K.M.S. and S. Palanki (1996). On-line optimization of batch processes with nonlinear manipulated input. *Chemical Engineering Science* **51**(3), 449–459.

Rahman, S. and S. Palanki (1998). State feedback synthesis for on-line optimization of batch reactors with multiple manipulated inputs. *Computers and Chemical Engineering* **22**(10), September.

Visser, E., B. Srinivasan, S. Palanki and D. Bonvin (2000). Feedback-based implementation scheme for batch process optimization. *Journal of Process Control* **10**(5), 399–410.

Zhang, G.P. and S. Rohani (2003). On-line optimal control of a seeded batch cooling crystallizer. *Chemical Engineering Science* **58**(9), 1887–1896.

Zuo, K. and W.T. Wu (2000). Semi-realtime optimization and control of a fed-batch fermentation system. *Computers and Chemical Engineering* **24**(2), 1105–1109.

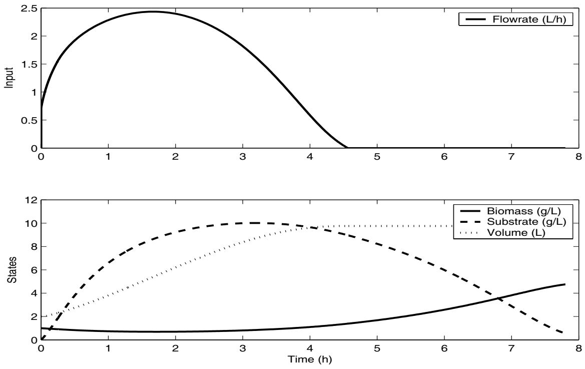

Fig. 1. Bioreactor Optimization Profiles