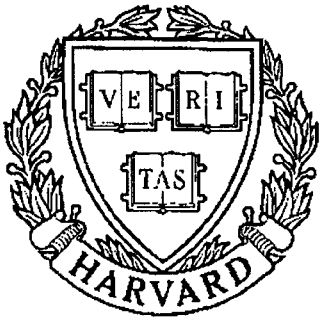
TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Optimal Feed Rate Profile Determination for Fed-Batch Fermentations in the Presence of Model-Plant Mismatch

by E. Zafiriou and J.M. Zhu

TR 89-48

Optimal Feed Rate Profile Determination for Fed-Batch Fermentations in the Presence of Model-Plant Mismatch*

Evanghelos Zafiriou† and Jian-min Zhu

Chemical Engineering
and Systems Research Center
University of Maryland
College Park, MD 20742

Abstract

Modelling error can be the cause of bad performance when optimal feed-rate profiles computed for a particular model are applied to the actual plant. This paper suggests the modification of the input trajectory from batch to batch, by using information from previous batches to modify the trajectories that are applied to the subsequent ones. The proposed approach does not required the remodeling of the process, but instead it redermines the input profile directly, so that a steady improvement is accomplished from batch to batch.

1 Motivation

Many industrial fermentations are operated in fed-batch mode. The problem of determining the optimal feed-rate profile is one that has attracted a lot of attention in the literature. This problem is usually addressed by defining an appropriate objective function and optimizing over the input profile. For simple models, this task can be carried out analytically. For more complex models, a numerical minimization [2] of the objective function is often performed [4]. Whichever approach is used to solve the optimal control problem, the fact is that the resulting input profile will be optimal for only the specific model (and model parameter values) used in the optimization. The mismatch between this model and the actual plant may result in very bad performance when the "optimal" profile is applied to the real plant. An approach that reduces the sensitivity to model uncertainty is to analytically obtain the optimal profiles as functions of the process states, so that the feed rate is implemented in a feedback model [3,6]. A limitation of this approach is that such results are limited to simple low-order models and require on line measurements of the state variables.

The approach proposed in this paper aims at using information from previous batches to improve the operation of the next. To do so, one could simply try to identify more accurately the model parameters and solve again the open-loop nonlinear optimal control problem between batches. This, however, is a very difficult and occasionally impossible task, because of errors in the structure of the often empirical models. Rather than attempt to go to the true optimum for the actual plant in one single step by remodelling and re-optimizing, the goal set in this paper is to gradually reach that optimum, by guaranteeing a steady performance improvement from batch to batch. Such a goal does not require the repeated remodelling of the process and seems to fit better the industrial environment. On the other hand it does not exclude such remodelling whenever possible.

*Supported in part by the National Science Foundations's Engineering Research Centers Program : NSFD CDR 8803012.

†To whom correspondence should be addressed.

2 Approach to the Problem

2.1 Numerical Solution of the Optimal Control Problem

Let us consider the general class of optimal control problems, finding $u(t)$ over the interval $[t_0, t_f]$ to minimize the performance index J:

$$\min_{u(t)} J = \phi(x(t_f)) \tag{1}$$

subject to

$$\dot{x} = f(x,u), \qquad x(t_0) = x_0 \tag{2}$$

as well as other appropriate constraints. Objective functions involving integrals from 0 to t_f can be converted into the above form by defining additional state variables.

The numerical optimization of (1) involves the selection of an initial profile $u_0(t)$ and iteration on the following step.

Step m: Obtain a search direction $s_m(t)$. Find

$$u_m(t) = u_{m-1}(t) + \alpha_m s_m(t) \tag{3}$$

either by using a fixed α_m or by carrying out a line search for $\alpha_m > 0$. Solve (2) for $u(t) = u_m(t)$ to obtain $x_m(t)$ and J_m. Go to Step (m+1).

The way $s_m(t)$ is obtained depends on the particular optimization method that is used. The computation might involve the use of the gradient g(t) of J with respect to u(t) (indirect search methods) or not (direct search methods).

In order to have an improvement in J after step m, the search direction must have the following descent property [7]:

$$< -g(t), s_m(t) > \quad > 0 \qquad (4)$$

Since fed-batch fermentations are usually linear in the substrate flowrate F(t), the use of F(t) as the control variable u(t), results in a singular control problem. To apply a numerical method to such a problem, one would have to express u(t) as [2]:

$$u(t) = c(t) + \sum_{i=0}^{M} h_i[1(t-t_i) - 1(t-t_{i+1})] \qquad (5)$$

where $c(t)$ is a continuous function, h_i is the height of the piecewise constant part of the input function in the interval $t_i < t < t_{i+1}$ and $1(t)$ is the unit step function. In this case the optimization is carried out over $c(t), h_0, \ldots, h_M, t_1, \ldots, t_M$. Then s(t) (and g(t)) has components $s^c(t), s_0^h, ..., s_M^h, s_1^t, ..., s_M^t$. In order to overcome computation difficulties, especially for high dimensional singular control problems, a number of techniques converting the singular control problem into a nonsigular one have been proposed. One way for fed-batch fermentations is to use the culture volume V(t) as the control variable u(t) [3]. This approach makes (1) nonsingular and u(t) a continuous function.

2.2 Analogy between Numerical Optimization and Plant Operation

The approach followed in this paper is to directly modify the input profile $u(t)$ during the course of successive batches. In doing so, we exploit the *analogy between the iterations during the numerical optimization of the appropriate cost functional (objective function) on one hand and the successive batches during the operation of the plant on the other.* More specifically, one way to solve the open-loop control problem is to numerically minimize the appropriate functional as described in Sec. 2.1. However, if this minimization is based solely on the process model, the resulting input profile will not be optimal for the true plant because of model-plant mismatch, as described above. We propose that this iteration includes information from the actual plant, *by corresponding each iteration to one of the successive batches.* This information will come from the measurement of appropriate variables during the batch. Note that since these measurements are not used for on-line control, where almost instantaneous measurements are required, we are not limited in their selection. For example, samples of the product can be gathered during the course of the batch and sent for analysis. The results will be used for the computation of a better $u(t)$ for the next batches. In this way, in every successive batch we will have a performance improvement and finally the input profile will converge to the "optimal" for the true plant.

The starting function $u(t)$ applied to the first batch could be the optimal for the model or some other reasonable good profile. Note, however, that the matter is somewhat more complicated than just described above, because the previous batches cannot necessarily provide all the information needed for the numerical algorithm. Consequently, the process model will be used to provide part of the information required to find the next $u(t)$. Significant model error could then cause a failure to converge to the true optimal profile. Conceptually, this bears a similarity to the Internal Model Control structure for continuous processing systems, where information both from the plant (measurement) and the parallel model is used in computing the next input. The algorithm used in the numerical minimization of the cost functional corresponds to the IMC controller. The same conceptual similarity exists also with the Operator Control theory for continuous processing systems [1], where algorithms for the numerical solution of operator equations are used as compared to the use of algorithms for the numerical optimization of a functional in this paper. Figure 1 gives a schematic representation of the analogy that was described in this section.

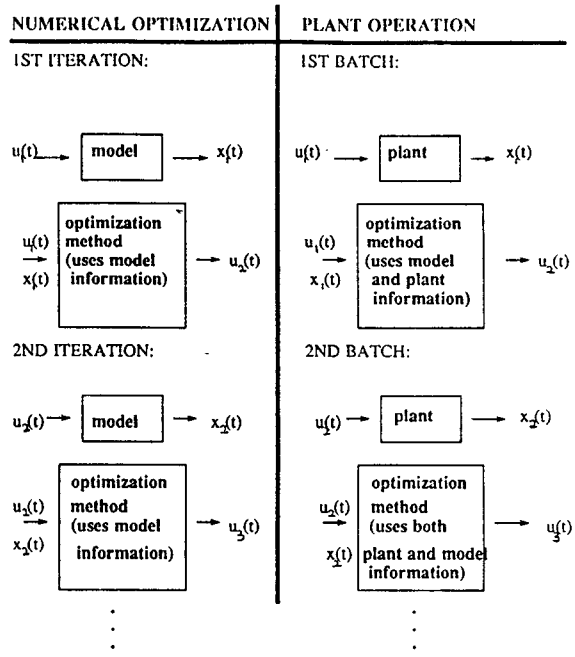


Figure 1: Analogy between the iterations of the numerical optimization and the successive batches during the plant operation.

3 Method Description

3.1 Indirect Search

Let us now consider the type of information required by an optimization algorithm that solves (1). When a gradient based method, like Steepest Descent or the Conjugate Gradient is used, the algorithm requires the knowledge of the gradient of the cost functional at the past input function $u_m(t)$. This computation involves two integrations. The first is forward integration of the differential equations describing the plant

$$\dot{x} = f(x, u_m) \qquad (6)$$

in order to obtain $x_m(t)$, where to simplify the notation we assume that all of the variables in the vector x appear in the cost functional and they can either be measured or estimated from measurements, speed of measurement not being important. The values of $u_m(t)$ and $x_m(t)$ are then used to compute the derivatives of $f(x,u)$ along these trajectories. The second integration is the backward integration of

$$\dot{\lambda} = -f_x^T \lambda, \qquad \lambda(t_f) = \nabla_x \phi(x(t_f,u)) \tag{7}$$

where T means transpose, f_x denotes the derivative of f with respect to x and ϕ is the objective function. The gradient is then obtained from $\lambda(t)$ and f_u (the derivative of f with respect to u). For the case of the singular control problem, the expression is given by [2]:

$$g(c(t),t_1,...,t_M,h_0,...,h_M) = \begin{bmatrix} f_u^T\lambda(t), t \in [t_0,t_f] \\ (f|_{t_1-} - f|_{t_1+})^T\lambda(t_1) \\ \vdots \\ (f|_{t_M-} - f|_{t_M+})^T\lambda(t_M) \\ \int_{t_0}^{t_1} f_u^T\lambda(t)dt \\ \vdots \\ \int_{t_M}^{t_f} f_u^T\lambda(t)dt \end{bmatrix} \tag{8}$$

If the non-singular approach is used (5), then $V(t) = u(t) = c(t)$ and the first component of (8) is used.

$$g(t) = f_u^T\lambda(t), \qquad t \in [t_0,t_f] \tag{9}$$

In the approach described in Section 2.2, the first forward integration is *not* carried out numerically, but rather *it is carried by the actual plant itself.* Its result $x_m(t)$ is the result of the previous batch. The remaining computations however require to compute f_x, f_u, at the trajectories of the previous batch $x_m(t)$, $u_m(t)$. This is accomplished by using the process model. Note that f_x, f_u define a linear time-varying system, obtained by linearization of the nonlinear model (6) along the trajectories $x_m(t)$, $u_m(t)$:

$$\dot{\xi} = f_x(t)\xi(t) + f_u(t)v(t) \tag{10}$$

The analogy between the computation of the gradient for numerical optimization, based on the model, and its computation during the plant operation, based both on model and plant operation, is shown schematically in Figure 2.

The result of the procedure described in the right-hand-side of Fig. 2 is an approximation of the real gradient, which would correspond to the use of the true f_x, f_u in (10). Since these are not available, we can only obtain an estimate. Again, note that since integration of (6) is carried out by the actual plant and u_m, x_m correspond to real data, both plant and model information is utilized In getting the gradient approximation. Having an estimate for the gradient g(t), a search direction can be obtained according to the particular indirect search method used.

3.2 Direct Search

An alternative to the use of an indirect search method is

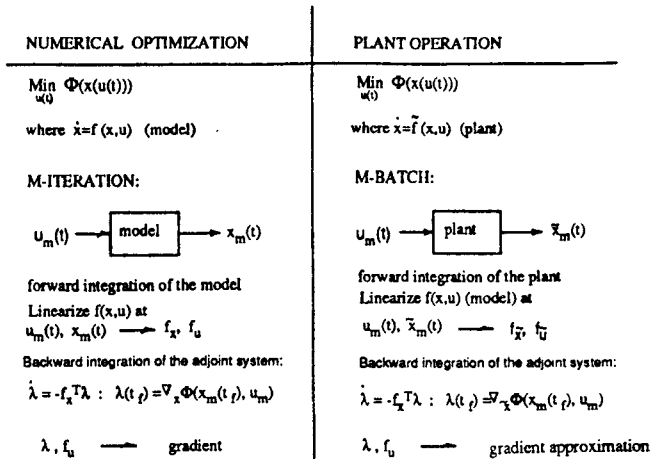


Figure 2: Gradient Computation

the use of direct search methods. These methods use only functional evaluations (J) and do not require the computation of the gradient g(t). Hence, there is no need to use f_x, f_u and so the model is not needed for the computations. Evaluations of J for different $u(t)$s can be obtained by applying u(t) into the actual plant.

The disadvantage of a direct search method is that there is no guarantee of a descent property and as a result, one may have to accept a temporary deterioration of performance for certain batches. The advantage is that they can be used for system where no or very poor models are available.

4 Robustness

The advantage of using a gradient-based method is obtaining a search direction s(t) with a descent property so that an improvement from batch to batch is obtained. In order to quantify the effect of model uncertainty on the convergence (from batch to batch), it is necessary to investigate the robustness of the proposed approach. The result of the procedure of Sec. 3.1 will be an estimate for the gradient g(t), which is then used to computer the search direction that will result in the next input profile $u_{m+1}(t)$. For example, if the Steepest Descent method is used, with constraints if necessary (Gradient Projection method), the search direction $s(t) = -Pg(t)$, where P is the projection matrix, and in order to have a descent property, we need to have:

$$< -g_{plant}(t), s(t) > \quad > 0 \tag{11}$$

where $g_{plant}(t)$ would be the gradient if the true f_x and f_u could be used in (7). Since this is not possible, because the plant is not known, a robustness condition for convergence from batch to batch can be obtained by requiring that the above condition be satisfied for the infimum of the LHS over all possible parameter values in the f_x, f_u used in computing $g_{plant}(t)$.

$$\inf_{parameters} < -g_{plant}(t), s(t) > \quad > 0 \qquad (12)$$

Satisfaction of (12) guarantees robustness in the sense that the approach of Sec. 3.1 will result in a performance improvement in every batch regardless of the fact that there is a model-plant mismatch.

5 Research Issues

In addition to model uncertainty, the measurement error will also result in errors in f_x, f_u. Similar conditions can be readily developed to address the problem of measurement error. The required computations, though done off-line, can be very extensive, if made in an ad hoc way. For this reason, we are currently trying to take advantage of the fact that the system in (10) is linear , though time-varying, by using the block-pulse function approximations and the structured singular value to carry out the computations.

Finally, it should be pointed out that the proposed method is really complementary to other existing approaches. For example, if possible, periodic remodeling can be carried out. This will help reduce the model uncertainty and therefore improve the method. However, such complex remodeling through nonlinear parameter identification is not required, since the approach taken is to simply improve the performance from batch to batch rather than try to get the best possible in only one step.

References

[1] C. Economou, *An Operator Theory Approach to Nonlinear Controller Design*, Ph.D. Thesis, California Institute of Technology, 1985.

[2] L. Hasdorff, *Gradient Optimization and Nonlinear Control*, John Wiley and Sons: New York, 1976.

[3] J. M. Modak and H. C. Lim, "Simple Nonsingular Control Approach to Fed-Batch Fermentation Optimization", Biotechnol. Bioeng., **33**, pp. 11-15, 1989.

[4] J. M. Modak and H. C. Lim, "Feedback Optimization of Fed-Batch Fermentation" , Biotechnol. Bioeng., **30**, pp. 528-540, 1987.

[5] S. J. Parulekar, J. M. Modak and H. C. Lim, "Optimal Control of Fed-Batch Bioreactors", Proc. of the Amer. Control Conf., p. 849, Boston, MA, 1985.

[6] G. Zoutendijk, *Mathematical Programming Methods*, North- Holland, 1976.