

SSQP for the solution of large-scale dynamic-economic optimization problems

J.J. van der Schot¹, R.L. Tousain¹, A.C.P.M. Backx² and O.H. Bosgra¹

¹ Mechanical Engineering Systems and Control Group

Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.

R.L.Tousain@wbmt.tudelft.nl

²IPCOS Technology b.v., Bosscheweg 145^a, 5282 WV Boxtel, The Netherlands.

Abstract

Dynamic optimization of industrial processes modeled by complex differential-algebraic equation systems is still a major challenge from an algorithmic point of view. We consider a special class of these problems, in which the objective is economic. The severe nonlinearity of this objective function relative to the smooth process behavior motivates the development of a new optimization algorithm called Successive Sequential Quadratic Programming (SSQP), which manages to drastically reduce the number of model integrations required to reach an optimum operating trajectory. The algorithm solved a 3800-variable (100 states) optimal grade transition problem for a HDPE reactor in less than 25 CPU minutes on a personal computer.

Keywords : dynamic optimization, economic optimization, algorithms, SQP, SSQP, HDPE.

INTRODUCTION

Economic optimization of the operation of industrial processes is believed to lead to significant increase in plant profitability. Whereas traditional optimization settings consider the problem of finding the best static operating conditions (e.g. Bailey *et al.*, 1993), attention is now shifting to economic operating optimality during *transients*: e.g. batch operation or grade transitions in continuous settings. The complexity of defining and finding optimal operating *trajectories* (over time) instead of optimal operating *points* (steady-state) has so far discouraged large-scale application of such technologies. Solution methods for large-scale dynamic optimization can be categorized in two main classes. In the *simultaneous* approach, both input and state trajectories are discretized and the constraints are enforced via collocation (Biegler, 1984). In the *sequential* approach, which is easier to implement, only the controls are discretized (parametrized) and the model equations are integrated and thus satisfied every iteration (for an overview see: Støren and Herzberg, 1995). The key in minimizing the calculation time for the sequential approach is to reduce the number of (expensive!) DAE integrations, e.g. by smartly exploiting a specific problem structure. In this article, we present the early results of a new sequential optimization method that has been developed specifically for dynamic-economic optimization problems on a medium to large scale.

PROBLEM DEFINITION

The kind of optimization problems considered in this work concentrates on the *economic* relevance of process *dynamics*. In this we seek to blend the

functionality of current generation economic optimizers (which do not consider process dynamics) with state-of-the-art technology from the field of dynamic optimization (which, however, does not usually concentrate on economic problems, but rather on control-oriented problems). To this end, we develop a dynamic ‘dollar conservation law’ and write the objective function V as a sum of profit made along the trajectory and a capital inventory term (Van der Schot, 1998):

$$\max V = \max \int_{t_0}^{t_f} \$_{revenues}(t) - \$_{expenses}(t) dt + \$_{inventory} \Big|_{t_0}^{t_f}$$

This comes down to maximizing the *added value* of the process over a fixed time interval $[t_0, t_f]$. In this, we define the dollar flows (\$/hr) as the product of physical flows (kg/hr) and product prices (\$/kg). These prices will on turn depend on product quality (e.g. Figure 2), which is given by simulation of a rigorous dynamic chemical process model. This model is included in the overall optimization as a set of equality constraints. We so obtain the general form of the (infinite dimensional) dynamic optimization problem:

$$\begin{aligned} \max_{u(t)} \quad & V = \int_0^{t_f} \Phi(x(t), u(t)) dt + \Psi(x(t_f), u(t_f)) \\ \text{s.t.} \quad & 0 = f(\dot{x}(t), x(t), u(t)) \\ & 0 \leq c(x(t), u(t)) \end{aligned} \quad (1)$$

with u inputs, x the state (and algebraic) variables, f the DAE model equations, c a general inequality constraint function and V the objective function. Φ and Ψ are functions that map the physical process variables $x(t)$ and $u(t)$ to dollars.

Table 1: Comparison between SQP and SSQP

SQP	SSQP
1. linearize $x(p)$ linearize $V(x)$ update Hessian of $V(x(p))$ 2. solve QP within TR $V(x(p))$ 3. evaluate V on nonlinear model V better : adopt new & go to 1 V worse : reduce TR & go to 2	1. linearize $x(p)$ do not approximate $V(x)$ 2. solve NLP within TR $x(p)$ 3. evaluate V on nonlinear model V better : adopt new & go to 1 V worse : reduce TR & go to 2

STANDARD APPROACH USING SQP

For large-scale settings, dynamic optimization problems such as (1) are usually discretized to yield a Non-Linear Program (NLP), which allows solution by a standard tool like Sequential Quadratic Programming (SQP). A reliable way to perform this discretization is the control parameterization method (e.g. Vassiliadis, 1993). With this method, the set of input signals $u(t,p)$ is discretized using a vector of parameters p , which become the decision variables for the optimizer. The model equations are integrated by an external simulation program, yielding a state profile $x(t,p)$. The objective function V and the constraints c are evaluated on this state profile and only these values are fed back to the optimizer, which tries to solve :

$$\begin{aligned} \max_p \quad & V = V(x(t, p)) \\ \text{s.t.} \quad & 0 \leq c(x(t, p)) \end{aligned}$$

where, for notational simplicity and without loss of generality, we suppose the objective and constraint functions to depend on the states $x(t)$ only. Because of the fact that all dynamic calculations are performed separately within an external simulator, this approach to solving (1) is also called *embedded model optimization*.

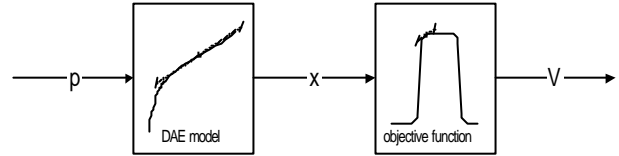
Most optimizers need gradient information which can be quite efficiently obtained using sensitivity equations (Støren and Herzberg, 1995). With this gradient information and an approximated Hessian, a Quadratic Program (QP) is set up and solved to yield a search direction, followed by a line search in that direction. Alternatively, the trust region (TR) approach to SQP sets up a QP within certain limits, i.e. it determines the step size *a priori* to be restrained to some surrounding of the current point (trajectory) in which we “trust” the quadratic model (Moré, 1983). The QP solution then yields a direction and step size which is implemented on the nonlinear model. If satisfactory improvement is obtained then the new point is adopted, otherwise the TR is reduced and a new QP solved.

The main problem with this classical optimization method is the cost of integrating the process model equations at each iteration. If strong nonlinearity is present then a gradient-based optimizer will need many iterations to converge because the linear (or quadratic) fit at each iteration varies greatly over the

search space. This combination leads to long calculation times for industrial scale problems, which impedes the move towards on-line use of dynamic optimization technology.

NEW APPROACH USING SUCCESSIVE SQP (SSQP)

We propose a refined approach to dynamic optimization, based on the observation that the objective function $V(x)$ may be much more strongly nonlinear than the process model $x(p)$. This presumed strong nonlinearity in the objective is certainly present if $V(x)$ is an economic objective function, because of sharp edges in pricing between on- and



off-spec product. At the same time, the relevant model equations are usually quite ‘smooth’ over some region around a given trajectory. This means that the trust region for a gradient (linear approximation) of $x(p)$ is much larger than the trust region for a gradient of $V(x)$ (Figure 1). Paradoxically though, the cost of evaluating $V(x)$ — a static function — is negligible compared to the evaluation of $x(p)$, which requires a complex DAE integration.

Figure 1 : gradient Trust Regions (dashed) of models (solid) for DAE system and objective function

If one treats this as an integrated problem, then the trust region of a gradient of $V(x(p))$ combined is approximately equal to the smallest of the two trust regions. This means that we must set up and solve many QPs to converge. However, if we separate the problem, then we can make use of the large trust region on the DAE model to limit the number of model integrations needed.

Starting from a discretized version of the sensitivity equations around an initial guess $x(p_o)$, we calculate the gradient dx/dp at a limited set of sampling times (Nevistic, 1997; Biegler *et al.*, 1990) for a limited set of variables (only those relevant to V or c). We use this gradient to construct an approximate linear affine ‘process model’. However, we do not approximate

$V(x)$. Instead, we solve an inner loop problem of the form:

$$\begin{aligned} \max_{\Delta p} V &= V(x(p_0) + \Delta p * dx / dp) \\ \text{s.t. } 0 &\leq c(x(p_0) + \Delta p * dx / dp) \\ \Delta p &\leq TR \end{aligned}$$

which defines an NLP (instead of a QP in the classic algorithm) for which the search space $\mathbf{D}p$ is confined to the trust region of dx/dp around $x(p_0)$. Using SQP, this inner loop problem is solved very rapidly since the evaluation of $\mathbf{D}p * dx/dp$ is only a linear operation instead of a full DAE integration. The solution of this subproblem is fed to the original nonlinear DAE model and the candidate trajectory is either accepted or rejected. The cycle then restarts until convergence is detected. The optimization sequence is summarized in Table 1.

EXAMPLE ON HDPE PROCESS

The proposed method has been tested on a medium-scale model of a high-density polyethylene (HDPE) reactor, functionally similar to the problem studied by McAuley (1992). This sample process has been chosen because polyethylene manufacturers are faced with an increasing need to operate flexibly with respect to different quality grades of polymer, which are mainly characterized by the density and so-called meltindex of the product. This flexibility calls for frequent grade transitions from one product to the other, during which the plant produces off-spec material which can only be sold at a lower price than on-spec product (Fig. 2). We have applied the methods developed in this work to define and find the economically optimal grade transition between two given grades (A and B). Finding such trajectories leads to improved overall plant profitability.

Test setup

The process model is a complex, stiff, nonlinear DAE system with 3800 variables (of which 100 states) implemented in SpeedUp. The optimization routine was developed in Matlab, with a simple communication protocol for the transfer of data and commands. As with all control parameterization methods, there was no need to alter the existing process model, hence numerical integration problems were avoided. All solution times reported were obtained on a multi-platform setup of a 500 MHz Alpha station (simulation code) and a Dual Pentium 200 MHz machine (optimizer code) and exclude communication time.

The manipulated variables chosen were those that most influenced the product quality and the inventory control. The optimizer used five input signals parameterized by 35 variables using piece-wise linear functions. The inputs were allowed to vary from time 0 to 12 and held constant at the (supposed known) desired steady-state values from $t=12$ onward in order to force the process to steady state within the interval

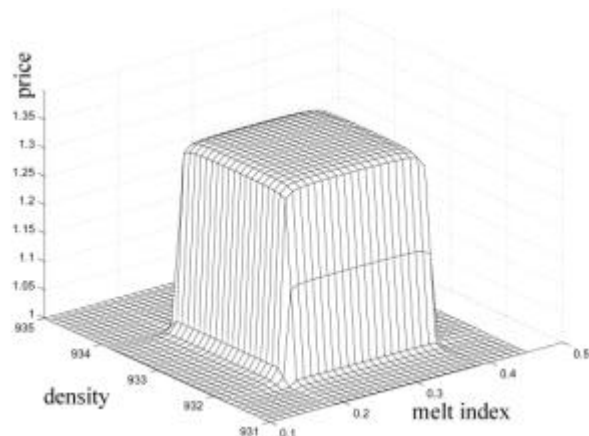


Figure 2: pricing of polymer as a function of 2 quality parameters (MI and dens).

concerned. This is similar to the use of control and prediction horizons in Model Predictive Control algorithms (Qin and Badgwell, 1996). The initial grade change trajectories were chosen as ramps, yielding a transition following the dashed line in Figure 3.

Solution

Maximizing the added value over a grade transition for this reactor poses a dynamic optimization problem that possesses the structure of Figure 1: a relatively smooth process model but sharp edges in the objective function. Therefore, the SSQP method was used to solve the problem.

The solution time for one of the runs performed is summarized in Table 2, where the different steps in the first column correspond to the steps in Table 1. The second column states how often the calculation step was performed, and the third column shows the total time spent on the step.

Table 2: Time logging for SSQP optimization

SSQP step	Number	Time(s)
Sensitivity equations	7	546
Solve NLP: evaluate V(x)	585	} 194
Solve NLP: evaluate dV/dx	215	
Simulations	16	667

The time needed for this optimization run was just under 25 CPU minutes. As we can see, the number of process simulations needed by the SSQP method is surprisingly low; only 16 simulations were carried out to converge to a (locally) optimum trajectory. This should be seen in contrast to the large number of inner loop iterations of 585. Comparison with standard SQP solution methods indicate an estimated improvement of about a factor 10 in solution time for the problem stated.

Note that although the number of outer loop iterations required by SSQP is limited, process model integration and sensitivity calculations still account for more than 85% of total optimization time. This figure shows — in line with our assumptions — that

integration of the process model equations is and remains the main bottleneck in dynamic optimization. Hence any attempt (such as SSQP) at reducing the number of model integrations is likely to reduce total optimization time.

Results on HDPE reactor

The optimization problem has been solved for several market conditions, each yielding a different grade-change policy. The results on the HDPE process show improved profitability of plant operation during a grade transition. The optimizer changed the input profiles to such an extent that both the off-spec time and the off-spec volume were reduced, which yielded an increase in added value (profit + net inventory) over the fixed interval of 25 hours. In the figure shown (3), the off-spec time was brought from 20.5 hours to 5 hours, whereas the volume was reduced from 70 to 12 tons. This yielded a profit increase of approximately \$10,000 over the entire grade transition with respect to the base case considered.

Figure 3: Initial (dashed) and optimized (solid) grade transition. Dotted: grade specification limits

CONCLUSIONS

The proposed SSQP method solves an NLP instead of a QP in the inner loop of the sequential approach towards dynamic optimization. This NLP is constructed from a linearized process model and an exact (non-approximated) objective function. For the class of (dynamic-economic) problems where the nonlinearity of the objective function is more severe than that of the process model, this yields much more accurate steps in the outer loop, hence reducing the number of nonlinear model integrations needed. The extra inner loop iterations incurred are inexpensive, thus total optimization time is reduced by an order of magnitude. SSQP is found to be efficient in solving the optimal grade transition problem for a realistic model of a HDPE reactor.

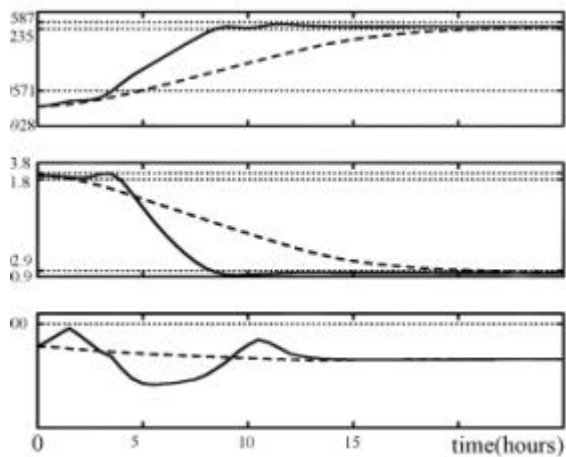
Generalization

Although SSQP has been developed specifically for dynamic-economic problems where the main nonlinearity resides in the objective function, the approach may be generalized to an arbitrary split up of a problem into a complex but meaningfully linearizable part (process model) and a simple but highly nonlinear part (objective). The leading idea should be to perform expensive accurate calculations only when necessary, and replace them by cheap approximations whenever possible. Other authors, such as Perregaard and Sørensen (1992) and Mujtaba and Machietto (1997) have developed ideas in the same direction, but no general two-stage strategy has yet been developed. We believe that SSQP provides a framework to work towards more efficient dynamic optimization, opening the way to on-line use.

NOTATIONS

N.B. many symbols in this article are used both as functions and as values. E.g. $V = V(x)$.

V	economic objective function
F	dollar profit function
Y	dollar inventory function
$\$$	money flow



c constraint function
 f DAE process model equations
 u input (manipulated) variables
 x state and algebraic variables
 p optimization parameters
 dx/dp gradient of x with respect to p

REFERENCES

- Bailey, J.K. and Hrymak, A.N. and Treiber, S.S. and Hawkins, R.B., 1993, Nonlinear optimization of a hydrocracker fractionation plant, *Comp. Chem. Eng.*, **17**, pp. 123-138.
- Biegler, L.T., 1984, Solution of dynamic optimization problem by successive quadratic programming and orthogonal collocation, *Comp. Chem. Eng.*, **8**, pp. 243-248.
- Li, W.C., Biegler, L.T. Economou, C.G. and Morari, M., 1990, A constrained pseudo-Newton control strategy for nonlinear systems, *Comp. Chem. Eng.* **14**, pp. 451-468.
- Moré, J.J., 1983, Recent developments in algorithms and software for trust region methods, in: *Mathematical Programming: the state of the art*, A. Bachem, M. Grötschel and B. Korte (eds). Berlin, 1983, Springer Verlag, pp. 258-287.
- Mujtaba, Iqbal M. and Macchietto, Sandro, 1997, Efficient optimization of batch distillation with chemical reaction using polynomial curve fitting techniques, *Ind. Eng. Chem. Res.*, **36**, pp. 2287-2295.
- Perregaard, J. and Sørensen, E.L., 1992, Simulation and optimization of chemical processes: numerical and computational aspects, *Comp. Chem. Eng.*, **16** (Suppl.), pp. S247-254
- Støren, S. and Herzberg, T., 1995, The sequential linear quadratic programming algorithm for solving dynamic optimization problems - a review, *Comp. Chem. Eng.*, **19** (Suppl.), pp. S495-S500.
- Qin, Joe S. and Badgwell, Thomas A., 1996, An overview of industrial Model Predictive Control Technology. in: *Preprints CPC-V, Lake Tahoe, USA*.
- Van der Schot, J.J., 1998, A practical approach to dynamic-economic optimization and control of industrial processes, M.Sc. report WBMT/ME-

S&C/A886, Delft University of Technology, Netherlands.

Vassiliadis, V., 1993, *Computational solution of dynamic optimization problems with general differential-algebraic constraints*, Ph.D. dissertation, Imperial College of Science, London.