# DIRECT SEQUENTIAL DYNAMIC OPTIMIZATION WITH AUTOMATIC SWITCHING STRUCTURE DETECTION

**Martin Schlegel**, **Wolfgang Marquardt** [1]

*Lehrstuhl für Prozesstechnik, RWTH Aachen University*
*D–52056 Aachen, Germany*

Abstract: In this paper we present a novel method for the numerical solution of dynamic optimization problems. After obtaining a first solution at a coarse resolution of the control profiles with a direct sequential approach, the structure of the control profiles is analyzed for possible switching times and arcs. Subsequently, the problem is reformulated and solved as a multi-stage problem, with each stage corresponding to a potential arc. Order and resolution of the control parameterization are adapted to the type of the particular arc. With a case study we show that accurate solutions with only few degrees of freedom can be obtained.

Keywords: optimal control, dynamic optimization, sequential approach, solution structure, switching times, arcs, multi-stage problem

## 1. INTRODUCTION

The optimization of the operation of batch processes or transient phases in continuous processes for either off-line or on-line applications requires the solution of dynamic optimization problems. It is still a challenge to obtain a high-quality solution for such problems efficiently, especially when the problem formulation contains large-scale models, e.g. those stemming from industrial applications. However, even for problems with just small models where the computation time is not an issue, the solution quality which can be obtained by numerical methods is not always satisfactory.

One important reason for this is that the analytical solution of a dynamic optimization problem consists of one or more intervals, the so-called *arcs* (Bryson and Ho, 1975). The control variables to be optimized are continuous and differentiable within each interval, but can jump from one interval to the next at the so-called *switching times*. This may pose problems to numerical solution methods, because the quality of the solution depends on the chosen parameterization order and resolution of the control variables. The solution quality can be insufficient, if the parameterization does not properly reflect the switching structure. However, the sequence and nature of the arcs is typically not known beforehand.

For practical applications, these problems are usually treated by either a) accepting the sometimes limited accuracy of the numerical solution, or by b) trying to interpret the numerical solution and to subsequently improve the solution. The latter approach typically involves human interaction such as visual inspection of the numerical solutions for finding the type and sequence of the arcs and often also requires the use of physical insight.

To the authors' knowledge there is only one attempt to try to determine the sequence of arcs in the solution structure automatically. Winderl and Büskens (2002) mention an algorithm for this purpose, though no details are presented. Some

[1] Corresponding author: marquardt@lpt.rwth-aachen.de

solution methods like the one of Vassiliadis *et al.* (1994) allow free interval lengths for control profile discretization, but the switching structure is not explicitly considered there.

In this paper we present the combination of a direct single-shooting approach with an automatic structure detection method. This procedure allows to solve a dynamic optimization problem with a control variable parameterization specifically tailored to reflect the sequence and nature of the different arcs, leading to a high solution quality involving only relatively few degrees of freedom. This is accomplished by a reformulation and subsequent solution of the problem as a *multi-stage problem*, where each stage corresponds to an arc of the solution structure as determined by the detection algorithm. This way a robust and efficient solution technique can be obtained.

## 2. PRELIMINARIES

### *2.1 Problem formulation*

We consider an optimal control problem of the following form:

$$\min_{\boldsymbol{u}(t),t_f} \Phi\left(\boldsymbol{x}\left(t_f\right)\right) \tag{P1}$$

$$\begin{aligned}
\text{s.t.}\quad \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)\,, \quad t\in[t_0,t_f]\,, && (1)\\
\boldsymbol{0} &= \boldsymbol{x}(t_0)-\boldsymbol{x}_0\,, && (2)\\
0 &\geq \boldsymbol{h}^x(\boldsymbol{x},t), \quad t\in[t_0,t_f]\,, && (3)\\
0 &\geq \boldsymbol{h}^u(\boldsymbol{u},t), \quad t\in[t_0,t_f]\,, && (4)\\
0 &\geq \boldsymbol{e}\left(\boldsymbol{x}\left(t_f\right)\right)\,. && (5)
\end{aligned}$$

In this formulation, $\boldsymbol{x}(t)\in\mathbb{R}^{n_x}$ denotes the vector of state variables with the given initial conditions $\boldsymbol{x}_0$. The process model (1) is formulated in form of the vector function $\boldsymbol{f}$. The time-dependent control variables $\boldsymbol{u}(t)\in\mathbb{R}^{n_u}$ are the degrees of freedom for the optimization. The final time $t_f$ can be either fixed or an unknown decision variable, as well. The objective function $\Phi$ is formulated as a terminal cost criterion for simplicity. Note that the more general formulation of an integral cost term can be easily converted into the above formulation. Furthermore, path constraints on the states (3) and control variables (4) and endpoint constraints on the state variables (5) can be employed. We assume that each constraint is formulated in terms of simple (lower and upper) bounds on the specific variables. More complex constraints and combined state and control path constraints can be converted into this formulation by adding additional equations to the model.

### *2.2 Necessary conditions of optimality*

By employing Pontryagin's Minimum Principle (Bryson and Ho, 1975), problem (P1) can be reformulated by introducing the *Hamiltonian* function $H(t)$ as

$$\min_{\boldsymbol{u}(t),t_f} H(t)=\boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})+\boldsymbol{\mu}^T\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \tag{P2}$$

$$\begin{aligned}
\text{s.t.}\quad \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)\,, \quad t\in[t_0,t_f]\,, && (6)\\
\boldsymbol{0} &= \boldsymbol{x}(t_0)-\boldsymbol{x}_0\,, && (7)\\
\dot{\boldsymbol{\lambda}}^T &= -\frac{\partial H}{\partial \boldsymbol{x}}\,, && (8)\\
\boldsymbol{0} &= \boldsymbol{\lambda}^T(t_f)-\left.\frac{\partial\Phi}{\partial\boldsymbol{x}}\right|_{t_f}-\boldsymbol{\nu}^T\left.\left(\frac{\partial\boldsymbol{e}}{\partial\boldsymbol{x}}\right)\right|_{t_f}\,, && (9)\\
0 &= \boldsymbol{\mu}^T\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u},t)\,, && (10)\\
0 &= \boldsymbol{\nu}^T\boldsymbol{e}(\boldsymbol{x}(t_f))\,. && (11)
\end{aligned}$$

Here, $\boldsymbol{\lambda}(t)\neq\boldsymbol{0}$ represents the vector of adjoint variables, $\boldsymbol{\mu}(t)\geq\boldsymbol{0}$ the vector of Lagrange multipliers for the path constraints and $\boldsymbol{\nu}\geq\boldsymbol{0}$ the vector of Lagrange multipliers for the terminal constraints. Note that the path constraints on states and control variables are combined into the vector function $\boldsymbol{h}=[\boldsymbol{h}^{xT},\boldsymbol{h}^{uT}]^T$. An optimal solution of problem (P2) fulfills the necessary conditions of optimality:

$$\begin{aligned}
\frac{\partial H(t)}{\partial\boldsymbol{u}} &= \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}}+\boldsymbol{\mu}^T\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{u}}=\boldsymbol{0}\,, && (12)\\
0 &= \boldsymbol{\mu}^T\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u},t)\,, && (13)\\
0 &= \boldsymbol{\nu}^T\boldsymbol{e}(\boldsymbol{x}(t_f))\,, && (14)\\
\mu_i &= 0,\text{ if } h_i<0;\ \mu_i>0,\text{ if } h_i=0\,, && (15)\\
\nu_i &= 0,\text{ if } e_i<0;\ \nu_i>0,\text{ if } e_i=0\,. && (16)
\end{aligned}$$

If a free final time is allowed, an additional transversality condition has to be also satisfied. The complementary conditions (15), (16) can be interpreted in a way that a specific Lagrange multiplier is positive if the corresponding constraint is active and zero otherwise.

### *2.3 Solution structure*

An optimal control profile $\boldsymbol{u}(t)$ consists of one or more arcs in such a way, that the control is continuous and differentiable within each arc, but may jump at the switching points. Conclusions about the possible solution structure can be derived from the necessary conditions (12)-(16). Equation (12) can be written separately for each control $u_i(t)$ as

$$\frac{\partial H(t)}{\partial u_i}=\boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}}{\partial u_i}+\boldsymbol{\mu}^T\frac{\partial\boldsymbol{h}}{\partial u_i}=0\,. \tag{17}$$

The expression consists of two parts, a system dependent part $\boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}}{\partial u_i}$ and a constraint dependent part $\boldsymbol{\mu}^T\frac{\partial\boldsymbol{h}}{\partial u_i}$. Assuming that we look at the solution structure in a given interval, we can distinguish two cases depending on the value of $\boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}}{\partial u_i}$:

(1) If $\boldsymbol{\lambda}^T\frac{\partial\boldsymbol{f}}{\partial u_i}\neq 0$, then it follows that $\boldsymbol{\mu}(t)\neq 0$ in order to fulfill (17). This implies that at least one of the path constraints $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u},t)$ must be active.

(2) If $\boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial u_i} = 0$, then $\boldsymbol{\mu}^T \frac{\partial \boldsymbol{h}}{\partial u_i} = 0$ must hold. Together with (15) this indicates that no path constraint can be active in that interval. The solution therefore lies inside the feasible region; it is not determined by the constraints. Such an interval is termed a *singular arc*.

The first case can be further distinguished in active control constraints or active state constraints. This leads to four different possibilities for the profile $u_i(t)$ in a specific interval, namely

(1) $u_i = u_{i,min}$ : the control is at its upper bound,
(2) $u_i = u_{i,max}$ : the control is at its lower bound,
(3) $u_i = u_{i,path}$ : the control is determined by an active state path constraint,
(4) $u_i = u_{i,sing}$ : the control is singular.

This distinction is useful for an automatic detection of the control structure.

## 3. NUMERICAL SOLUTION

Solution strategies for dynamic optimization problems of form (P1) can be classified into indirect methods, which use the first-order necessary conditions from Pontryagin's Minimum Principle (see Section 2.2) for a reformulation of the problem as a multi-point boundary value problem, and direct methods, which solve problem (P1) directly. We refer to Srinivasan *et al.* (2003) for a review of the various variants of these methods.

In this work, we use the control vector parameterization approach, also referred to as single-shooting or sequential approach, a direct method which solves the problem by conversion into a nonlinear programming problem (NLP) through discretization of the control variables $\boldsymbol{u}(t)$. It is important to note that our solution approach does not require an explicit derivation of the necessary conditions (12)-(16).They are just used for a theoretical justification of the method proposed later.

### *3.1 Control vector parameterization*

In the control vector parameterization approach (Kraft, 1985) only the control variables $\boldsymbol{u}(t)$ are discretized explicitly. The discretization parameters are the degrees of freedom for the optimization. The profiles for the state variables $\boldsymbol{x}(t)$ are obtained by forward numerical integration of the model (1) for a given input. For the parameterization of the control profiles $u_i(t)$ often piecewise polynomial approximations are applied. We use a B-spline representation for this purpose where the discretized control variables can be written as

$$u_i(t) = \sum_{j=1}^{n_i^u} \hat{u}_{i,j}\, \varphi_j^{(m)}\,, \tag{18}$$

with $n_i^u$ denoting the number of parameterization functions for the control variable $u_i$. Depending on the choice of the order $m$ of the B-spline function $\varphi^{(m)}$ different orders can be realized. Our solution framework uses piecewise constant ($m = 1$) and piecewise linear ($m = 2$) parameterizations. Let $\Delta_i$ denote the set of discretization time points for each control variable $u_i(t)$.

### *3.2 Reformulation as NLP problem*

Once the control profiles have been discretized, problem (P1) can be reformulated as an NLP:

$$\min_{\hat{\boldsymbol{u}}, t_f} \Phi = \Phi\left(\boldsymbol{x}\left(\hat{\boldsymbol{u}}, t_f\right)\right) \tag{P3}$$

$$\text{s.t.} \quad 0 \geq \boldsymbol{h}(\boldsymbol{x}, \hat{\boldsymbol{u}}, t_i), \quad \forall\, t_i \in \Delta\,, \tag{19}$$

$$0 \geq \boldsymbol{e}\left(\boldsymbol{x}\left(t_f\right)\right)\,. \tag{20}$$

The vector $\hat{\boldsymbol{u}}$ contains all discretization parameters of the control variables. For evaluation of the state path constraints in (19) we cannot directly use the continuous formulation of (P1), since the NLP requires a finite number of constraints. Therefore, all state path constraints are evaluated point-wise, for example on the unified mesh of all control variables $\Delta := \bigcup_{i=1}^{n_u} \Delta_i$. The number of mesh points contained in $\Delta$ is denoted with $n_\Delta$.

### *3.3 Solution of the NLP problem*

Once problem (P3) has been formulated, it can be solved by a suitable NLP solver for a given $\hat{\boldsymbol{u}}$. Typically, a sequential quadratic programming (SQP) method (Nocedal and Wright, 1999) is used for this purpose. A detailed discussion of NLP theory is beyond the scope of this paper. We just mention the key aspects which are required later.

An optimal solution of the NLP problem (P3) fulfills the Karush-Kuhn-Tucker conditions of optimality (Nocedal and Wright, 1999). These are formulated based on the *Lagrangian* function, which is defined as

$$L(\hat{\boldsymbol{u}}, t_f, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\nu}}) = \Phi(\boldsymbol{x}(\hat{\boldsymbol{u}}, t_f)) - \hat{\boldsymbol{\mu}}^T \boldsymbol{h}(\hat{\boldsymbol{u}}, \boldsymbol{x}, t) - \hat{\boldsymbol{\nu}}^T \boldsymbol{e}(\boldsymbol{x}(t_f))\,. \tag{21}$$

Each discrete constraint has an associated discrete Lagrange multiplier, $\hat{\mu}_i$ or $\hat{\nu}_i$, respectively. They are related to the Lagrange multipliers $\boldsymbol{\mu}(t)$ and $\boldsymbol{\nu}$ of the continuous problem (P2). The value of each of the discrete multipliers provides information about the status of the particular constraint at the optimal solution. Analogously to the continuous case, they fulfill complementary conditions. In the following we will make use of the convention, that $\hat{\mu}_i = 0$, if the corresponding constraint is not active, $\hat{\mu}_i > 0$, if it is at the upper bound and $\hat{\mu}_i < 0$, if it is at the lower bound.

### *3.4 Resolution of the control discretization*

Obviously, it is desirable to obtain a solution of the discretized problem (P3) which is close to the true solution of the optimal control problem (P1). However, a very fine discretization mesh $\Delta_i$ is not a favorable option for practical problems mainly because of three reasons: a) numerical accuracy, b) robustness and c) computational efficiency. If a control profile is represented by a very fine discretization of the form (18), the NLP solution algorithm has to deal with a large number of decision variables. However, as indicated before, typical control profiles may exhibit regions where there are no significant differences between the values of neighboring control vector parameters $\hat{u}_{i,j}$, for example in those parts of the control profile where it is governed by an active control path constraint. In these regions, a fine discretization would not be required to reflect the true solution. But also on singular arcs, a too fine parameterization can lead to numerical problems. Also, the computational effort required for the solution of a dynamic optimization problem with the sequential method is strongly correlated to the number of parameterization functions and renders a fine discretization unattractive from an efficiency perspective. If a-priori knowledge about the solution structure is available, it can be considered while setting up a possibly non-equidistant discretization mesh. The method suggested in the following does not require such an a-priori information. Instead, an appropriate parameterization is determined automatically.

## 4. AUTOMATIC STRUCTURE DETECTION

The proposed method consists of three main steps:

(1) a (probably coarse) solution of problem (P3),
(2) a detection of the arcs in the solution structure,
(3) a reformulation and solution of a multi-stage problem.

The multi-stage problem in step 3 treats each arc determined in step 2 as a separate stage, where the parameterization order (constant or linear) is chosen according to the type of the arc. For example, a stage corresponding to an arc with $u = u_{max}$ will be parameterized by piecewise constant functions, whereas a piecewise linear parameterization is used on a singular arc.

### *4.1 Structure detection algorithm*

The detection of the arcs and switching times in the control profiles obtained as a solution of problem (P3) works with the help of the concepts of optimal control theory which have been outlined in Sections 2.2 and 2.3. The solution of the discretized problem (P3) including the Lagrange parameters for the path constraints can be used for this purpose. In the following, the corresponding discrete Lagrange parameters for $h_i^x$ and $h_i^u$ are denoted with $\hat{\mu}_i^x$ and $\hat{\mu}_i^u$, respectively. For brevity we restrict ourselves to the case of a single control variable and a single state path constraint. Then, a simplified algorithm looks as follows:

**Algorithm 1** Structure detection

```
for i = 1, n_Δ do
  if μ̂^u(i) < 0 then
    utype(i) ← 'min'
  else if μ̂^u(i) > 0 then
    utype(i) ← 'max'
  else
    if μ̂^x(i) ≠ 0 then
      utype(i) ← 'path'
    else
      utype(i) ← 'sing'
    end if
  end if
end for
j ← 1, atype(1) ← utype(1)
for i = 1, n_Δ − 1 do
  if utype(i+1) ≠ utype(i) then
    j ← j + 1
    atype(j) ← utype(i)
  end if
end for
L ← j
```

As a result, we obtain the number of arcs $L$ and their type in the vector *atype*. *utype* denotes the type ($u_{max}$, $u_{sing}$, etc.) of the control variable in each discretization interval. Also, initial values for the lengths of the intervals and therefore the switching times can be extracted from the single-stage solution. This information is used for a subsequent reformulation as a multi-stage problem.

### *4.2 Multi-stage problem formulation*

The model $\boldsymbol{f}$ in problem (P1) is continuous and only comprises a single set of state variables $\boldsymbol{x}$ and controls $\boldsymbol{u}$. Such a model describes a process in a single mode or a single discrete state and can therefore be called a *single-stage model*. The corresponding optimization problem is a *single-stage problem*. In contrast, *multi-stage models* are used to describe situations, where the process consists of a sequence of modes or discrete states, with switching from one mode to the next at some time instant $t_k$ (Vassiliadis *et al.*, 1994).

In the context of this paper we explicitly only treat problems which only have one physical stage, i.e. single-stage problems. However, we can use the multi-stage formalism in order to obtain an alternative formulation and solution method for these problems by using the results of the structure detection algorithm.

We define a set $K = \{1, \ldots, L\}$ comprising the indices of all stages. A multi-stage reformulation of the original problem then can be written as

$$\min_{\boldsymbol{u}_k(t), \boldsymbol{x}_k(t_k), t_k} \Phi_L(\boldsymbol{x}_k(t_k)) \tag{P4}$$

$$\dot{\boldsymbol{x}}_k = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, t),\ t \in [t_{k-1}, t_k]\,, \forall\, k \in K, \tag{22}$$

$$\boldsymbol{0} = \boldsymbol{x}_1(t_0) - \boldsymbol{x}_0\,, \tag{23}$$

$$\boldsymbol{0} = \boldsymbol{x}_{k+1}(t_k) - \boldsymbol{x}_k(t_k)\,,\ \forall\, k \in K \tag{24}$$

$$0 \geq \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, t),\ t \in [t_{k-1}, t_k]\,, \forall\, k \in K, \tag{25}$$

$$0 \geq \boldsymbol{e_L}\left(\boldsymbol{x_L}\left(t_L\right)\right)\,. \tag{26}$$

The index $k$ denotes quantities belonging to stage $k$. The time indices are chosen such that e.g. the time horizon of stage 1 runs from $t_0$ to $t_1$. The final time of the last stage $t_L$ corresponds to the final time $t_f$ of the single-stage problem. The models $\boldsymbol{f}_k$, $k \in K$, are all identical copies of model (1). The initial conditions of the first stage ($k = 1$) are set to $\boldsymbol{x}_0$ by equation (23) and are therefore also the same as in equation (2). In addition, the multi-stage problem requires so-called stage transition or mapping conditions (24), which map the state variable values $\boldsymbol{x}_k$ across the stage boundaries. The objective function of a multi-stage problem is often obtained by summing up individual costs $\Phi_k(\boldsymbol{x}_k(t_k))$ formulated for each model stage $k \in K$. Here, just the objective function value at the endpoint of the last stage $\Phi_L(\boldsymbol{x}_L(t_L))$ has to be minimized. The influence of the other stages is propagating into the final stage via the initial condition $\boldsymbol{x}_L(t_{L-1})$.

The path constraints obviously should be fulfilled in all stages and are therefore also copied as in equation (25). Similarly to the objective function the endpoint constraints should be only evaluated at the final time point $t_L$. Therefore, they are formulated only for stage $L$ (equation (26)).

In the single-stage problem, the final time $t_f$ might be a degree of freedom. In the reformulated problem each stage corresponds to an arc of the solution structure. Since we know that the solution of the discretized problem (P3) is not likely to exactly match the switching times of the true solution, we would like to introduce those switching times as additional degrees of freedom. The multi-stage formulation provides a well-suited framework for this purpose, because the length of the arcs (which can be easily converted into the switching times) are just the parameters $t_k$. Therefore, these are included into the set of optimization degrees of freedom in problem (P4). Note that the times $t_k$ are even free in those cases where $t_f$ is fixed, though in principle one of them then can be determined from the others.

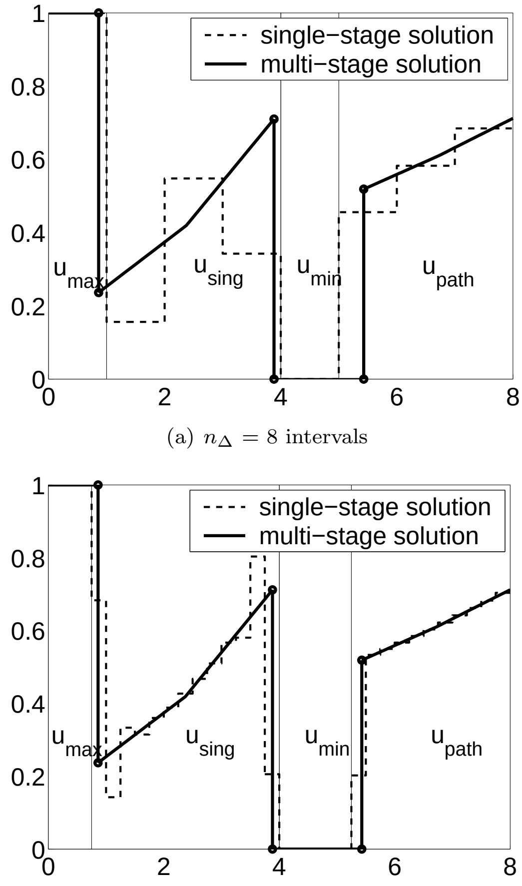
The control variables $\boldsymbol{u}(t)$ in the original problem are now also present in each model stage as $\boldsymbol{u}_k(t)$. In order to be able to fulfill the mapping conditions at the stage boundaries, the multi-stage optimization problem requires additional degrees of freedom: the initial values of the state variables $\boldsymbol{x}_k$ of all stages but the first. Those have to be determined by the optimization algorithm such that at the optimal point the conditions (24) are satisfied. As indicated earlier, the order $m$ of the B-splines used for discretization is adapted to the type of arc. More precisely, we use $m = 1$ and one interval, if atype = 'min' or 'max', and $m = 2$ and two discretization intervals for atype = 'path' or 'sing'.

The maybe coarse solution of the single-stage problem allows to derive good initial values for the optimization variables of (P4). A proper initialization of $\boldsymbol{u}_k(t)$ is available through interpolation of the single-stage solution in the particular region of the time horizon belonging to arc (or stage) $k$. The multi-stage problem is then solved in the same way as the single-stage problem, i.e. through conversion into an NLP problem as described above.

## 5. ILLUSTRATIVE EXAMPLE

The numerical concepts presented above have been implemented into the software tool DyOS (DyOS, 2002). The implementation allows optimization of any model (1) compliant to the Cape-Open ESO interface standard (Keeping and Pantelides, 2000), e.g. through the modeling and simulation package gPROMS (gPROMS, 2002). The user just has to specify the problem in form of (P1). Switching structure detection, reformulation and solution of the multi-stage problem are carried out automatically. The sample problem has been solved by using SNOPT (Gill *et al.*, 1998) as NLP solver and a modified version of LIMEX (Schlegel *et al.*, 2004) as numerical integrator.

As an example we consider the optimization of a fed-batch bioreactor with inhibition and a biomass constraint. Details of the problem can be found elsewhere (Srinivasan *et al.*, 2003). The problem has been solved for two different cases: starting from a single-stage solution with $n_\Delta = 8$ and $n_\Delta = 32$ equidistant discretization intervals. As the solution plots in Figure 1 show, the optimal profile of the substrate feed rate exhibits a complex switching structure with $L = 4$ arcs. The arcs of the true solution have been detected correctly. The reformulated multi-stage solution is of higher quality, because a) the switching times have been captured properly (which was not the case in the single-stage solutions) and b) the piecewise-linear parameterization in the singular and path-constrained regions provide good approximations of the true solution with very few degrees of freedom. Note that the multi-stage solutions derived from the two different single-stage solutions are essentially identical.




(a) $n_\Delta = 8$ intervals

(b) $n_\Delta = 32$ intervals

Fig. 1. Optimal solution profiles for $u$.

## 6. CONCLUSIONS

We have presented a direct method for the numerical solution of dynamic optimization problems, which determines and explicitly considers the structure of the optimal control profiles. The method yields a high solution accuracy with a small number of degrees of freedom. In contrast to indirect methods the method can be easily applied to large-scale problems. No a-priori knowledge about the solution structure is required.

The multi-stage solution can be useful in on-line applications like moving-horizon optimization, because taking the switching structure into account can add robustness and computational efficiency to the numerical solution method.

As presented here, the arcs in the solution structure can be only captured properly, if they are already present in the single-stage solution. This assumption might be violated in some cases, e.g. if the single-stage discretization is too coarse. A highly resolved solution of the initial problem would most likely contain all arcs, but cannot be solved efficiently. Current work is therefore aiming at including grid point adaptation techniques on both, the single and the multi-stage level in order to capture the correct solution structure efficiently also in complex problem. Recent extensions of the method allow to handle problems with more than one control variable, as well as enable an efficient treatment of large-scale process models.

## REFERENCES

Bryson, A.E. and Y.-C. Ho (1975). *Applied Optimal Control*. Taylor & Francis. Bristol, PA.

DyOS (2002). *DyOS User Manual, Release 2.1.* Lehrstuhl für Prozesstechnik, RWTH Aachen University. Aachen, Germany.

Gill, P.E., W. Murray and M.A. Saunders (1998). SNOPT: An SQP algorithm for large-scale constrained optimization. Technical report. Stanford University. Stanford, USA.

gPROMS (2002). *gPROMS User Guide (Release 2.1.1)*. Process Systems Enterprise Ltd.. London, UK.

Keeping, B. and C. Pantelides (2000). Numerical Solvers Open Interface Specification Draft. Technical Report CO-NUMR-EL-03. CPSE, Imperial College. London.

Kraft, D. (1985). On converting optimal control problems into nonlinear programming problems. *Comput. Math. Prog.* **15**, 261–280.

Nocedal, J. and S.J. Wright (1999). *Numerical Optimization*. Springer-Verlag.

Schlegel, M., W. Marquardt, R. Ehrig and U. Nowak (2004). Sensitivity analysis of linearly-implicit differential-algebraic systems by one-step extrapolation. *Appl. Num. Math.* **48**(1), 83–102.

Srinivasan, B., S. Palanki and D. Bonvin (2003). Dynamic optimization of batch processes I. Characterization of the nominal solution. *Comp. Chem. Eng.* **27**, 1–26.

Vassiliadis, V.S., R.W.H. Sargent and C.C. Pantelides (1994). Solution of a class of multistage dynamic optimization problems: 1. Problems without path constraints. *Ind. Eng. Chem. Res.* **33**(9), 2111–2122.

Winderl, S. and C. Büskens (2002). Exploiting the special structure for real-time control of optimal control problems with linear controls: Numerical experience. In: *Proc. Appl. Math. Mech.*. Vol. 1. Wiley. pp. 484–485.