ISBN 978-83-60810-51-4

Parallel Simultaneous Approach for optimal control of DAE systems

Paweł Drąg
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11/17, 50-372, Wrocław, Poland
Email: pawel.drag@pwr.wroc.pl

Krystyn Styczeń
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11/17, 50-372, Wrocław, Poland
Email: krystyn.styczen@pwr.wroc.pl

***Abstract*—In the paper two approaches of parallelization for solving optimal control problems of ODE and index-1 DAE systems were presented and discussed. DAE Optimization Problem can be treated by Direct Nonlinear Programming Approach in two manners, known as Sequential Approach and Simultaneous Approach. Simultaneous Approach seems to be more reliable, because provides initial states in periods and discretized control variables. Therefore there is a possibility of efficient use of Optimization with multiple shooting, which was developed to handle unstable DAE systems. In the article some parallelization methods of the Sequential Quadratic Programming were discussed and compared both in the Jacobian calculation and the numerical integration of DAE models. Augmented objective function, based on Mathematical Programming with Complementarity Constraints, was proposed. The illustrative simulations of Catalyst Mixing Problem were performed in MATLAB, which is a commonly known programming environment.**

***Index Terms*—optimal control, DAE systems, Simultaneous Approach**

I. Introduction

For last 30 years differential-algebraic systems have been attracted the attention of many scientists around the world. One of the first monographs justifies the use of such equations. "In the last decade the use of differential-algebraic equations (DAE's) has become standard modeling practice in many applications, such as constrained mechanics and chemical process simulation. The advantages of using implicit, often computer-generated models for dynamical processes are encouraging the use of DAE's in new areas" [6]. Then the study was conducted to highlight the basic types of systems and their properties [6], [9], [14], [16].

Today, practical use of the optimal control methods in real systems in the chemical and aerospace industry have been widely documented in [4], [5], [10].

One of the most popular approach for DAE Optimization Problems is Direct NLP Approach. At this moment one should decide, how many variables are there. It means, what is the scale of the problem. The second important aspect is, what is their origin and what the variables do actually mean. If the variables denote only discretized control function, it is reasonable to use Sequential Approach. If in addition there are variables connected with initial conditions of state trajectory, Simultaneous Approach can be used. In both mentioned situations, Multiple Shooting method is appropriate, especially when unstable DAE system is considered. So, "Optimization with multiple shooting serves as a brige between sequential and direct transcription approaches (...)" [5].

Advanced optimal control methods combine generalized mathematical models, elastic mode optimization algorithms, numerical analysis for solving DAE systems and high performance programming environments.

Some ideas from smooth optimization were adjusted to non-smooth problems and successfully used there. This applies, for example, Mathemmatical Programming with Complementarity Constraints, where the formulation of the objective function may mirror various specifc features of the system [1], [2], [17]. This dependence can also act in the opposite direction. So, the use of of the scalar product with the penalty coeffcient was proposed to take into account the equality constraints resulting from the multiple shooting method in the objective function.

This paper was organized as follows. First, the optimal control problem of DAE systems was stated. Then the elements of Simultaneous Approach were described. Our attention was focused on reliable using of both NLP and DAE solvers. Before numerical examples the discussion about parallelization methods was conducted and some comments about using Multi-Step Approach were made. Catalyst Mixing Problem is one of tasks, which can be arbitrarily large. Simulations were executed in many ways with different number of parallel processes.

II. Optimal control of DAE systems

In the paper the following DAE optimization problem was considered

$$\varphi(p) = \sum_{l=1}^{N_T} \Phi^l(y^l(t_l), z^l(t_l), p^l) \to min, \tag{1}$$

subject to

$$\frac{dy^l}{dt} = f^l(y^l(t), z^l(t), p^l) \tag{2}$$

with initial conditions

$$y^l(t_{l-1}) = y_0^l \tag{3}$$

and algebraic equations

$$g^l(y^l(t), z^l(t), p^l) = 0, t \in (t_{l-1}, t_l], l = 1, ..., N_T \tag{4}$$

with bounds

$$p_L^l \leq p^l \leq p_U^l, \tag{5}$$

$$y_L^l \leq y^l \leq y_U^l, \tag{6}$$

$$z_L^l \leq z^l \leq z_U^l \tag{7}$$

and constraints

$$h(p^1, ..., p^{N_T}, y_0^1, y^1(t_1), y_0^2, y^2(t_2), ..., y_0^{N_T}, y^{N_T}(t_{N_T})) = 0. \tag{8}$$

There are differential variables $y(t)$ and algebraic variables $z(t)$ in the DAE system in semiexplicit form. The assumption of invertibility of $g(-, z(t), -)$ permits an implicit elimination of the algebraic variables $z(t) = z[y(t), p]$. While there are N_T periods in DAE equations, time-dependent bounds or other path constraints on the state variables are no longer considered [5]. The control profiles are represented as parameterized functions with coefficients that determine the optimal profiles [20], [21]. The decision variables in DAE equations appear only in the time independent vector p^l. Algebraic constraints and terms in the objective function are applied only at the end of each period.

The problem (1)-(8) can be represented as the following nonlinear program

$$\phi(p) \to min \tag{9}$$

subject to

$$c_i(p) = 0, i \in E, \tag{10}$$

$$c_i(p) \leq 0, i \in I. \tag{11}$$

In the literature one can find a lot of methods solving the above problem (see [4], [5], [12], [15]).

III. Simultaneous Approach

The method described in this article is a part of the trend "Discretize then Optimize". This approach has the advantage of directly finding good approximate solutions, which are feasible to the state equations. From literature [5] it is known, that this formulation requires an accurate level of discretization of the control and state profiles.

In the considered approach, the time domain is partitioned into smaller time elements and the DAE models are integerated separately in each element [4], [5], [12], [13], [18].

Control variables are represented as piecewise polynomials [20], [21]. Optimization is performed with respect to polynomials coefficients. Sensitivities are obtained for both the control variables and the initial conditions of the states in each element. Equality constraints are added to the nonlinear program in order to link the elements and ensure that the states remain continous over time. Inequality constraints for states and control can be imposed directly at the grid points, although constraints on the state profiles may be violated between grid points [5].

There is a sketch of simultaneous dynamic optimization strategy on the fig. 1. This approach consists of three main elements. There are NLP solver, DAE solver and sensitivity calculations. All of this parts, especially from parallel calculations point of view, were described and commented below.

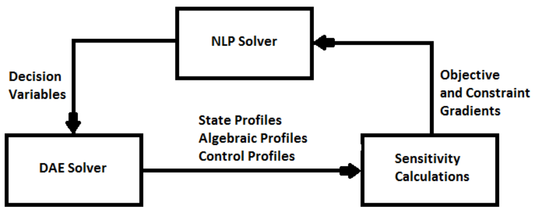

Figure 1. Sketch of simultaneous dynamic optimization strategy.

A. NLP solver

"Discretize then Optimize" approach enables us to use the full machinery of large-scale NLP solvers. Now SQP codes are considered. For large-scale problems the appropriate SQP codes are $filterSQP$ [7], $MUSCOD-II$ [12], [13] and $SOCS$ [4]. For nonlinear programming problems with less than a few hundreds of variables $fmincon$ solver can be treated as quite suitable [5].

$fmincon$ is coupled to the MATLAB environment and applies l_1 merit function line search and BFGS updates of the Hessian [5], [8].

Algorithm 1 presents steps executed by $fmincon$ [8], [15]. This algorithm has to be completed by two comments. The general nonlinear programming problem is modeled by linearization both the inequality and equality constraints to obtain

$$\widetilde{\phi}(d) = \phi_k + \nabla \phi_k^T d + \frac{1}{2} d^T \nabla_{pp}^2 \mathcal{L}_k d \to min \tag{12}$$

subject to

$$\nabla c_i(p_k)^T d + c_i(p_k) = 0, i \in E, \tag{13}$$

$$\nabla c_i(p_k)^T d + c_i(p_k) \leq 0, i \in I. \tag{14}$$

The second note is, that strategy for choosing μ in the Algorithm 1 considers the effect of the step on a model of the merit function, so μ has to satisfy the inequality

$$\mu \geq \frac{\nabla \phi_k^T d_k + \frac{\sigma}{2} d_k^T \nabla_{pp}^2 \mathcal{L}_k d_k}{(1-\rho) \parallel c_k \parallel_1}. \tag{15}$$

If the value of μ from previous iteration of the SQP method satisfies eq. (15), it is left unchanged. Otherwise, μ is increased so that satisfies this inequality with some margin. The constant σ is used to handle the case in which Hessian $\nabla_{xx}^2 \mathcal{L}_k$ is not positive definite. We define $\sigma = 1$, if $d_k^T \nabla_{xx}^2 \mathcal{L}_k d_k > 0$, and $\sigma = 0$ otherwise.

The l_1 merit function for the problem (9)-(10) takes the form

$$\Phi_1(p; \mu) = \phi(p) + \mu \|c(p)\|_1. \tag{16}$$

The directional derivative of Φ_1 in the direction d_k satisfies

$$D(\Phi_1(p_k; \mu); d_k) = \nabla \phi_k^T d_k - \mu \|c_k\|_1. \tag{17}$$

Algorithm 1 Line Search SQP Algorithm

```
    Choose parameters $\eta \in (0, 0.5)$, $\tau \in (0, 1)$, and an initial pair $(p_0, \lambda_0)$;
 2: Evaluate $\phi(p_0), \nabla\phi(p_0), c_i(p_0)$,
    $A_0 = [\nabla c_1(p_0), \nabla c_2(p_0), \dots, \nabla c_m(p_0)]^T$;
 4: If a quasi-Newton approximation is used, chose an initial $n \times n$ symmetric positive definite Hessian approximation $B_0$, otherwise compute $\nabla^2_{pp}\mathcal{L}_0$;
    **while** convergence test is not satisfied **do**
 6:     Compute $d_k$ by solving eq. (12)-(14); let $\lambda$ be the corresponding multiplier;
        Set $d_\lambda \leftarrow \widehat{\lambda} - \lambda_k$;
 8:     Choose $\mu_k$ to satisfy eq. (15) with $\sigma = 1$;
        Set $\alpha_k \leftarrow 1$;
10:     **while** $\Phi_1(p_k + \alpha_k d_k; \mu_k) > \Phi_1(p_k; \mu_k) + \eta\alpha_k D_1(\phi(p_k; \mu_k) d_k)$ **do**
            Reset $\alpha_k \leftarrow \tau_\alpha \alpha_k$ for some $\tau_\alpha \in (0, \tau]$;
12:     **end while**
        Set $p_{k+1} \leftarrow p_k + \alpha_k d_k$ and $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k d_\lambda$;
14:     Evaluate $\phi_{k+1}, \nabla\phi_{k+1}, c_{k+1}, A_{k+1}$ (and possibly $\nabla^2_{xx}\mathcal{L}$);
        **if** a quasi Newton approximation is used, set **then**
16:         Set $s_k \leftarrow \alpha_k d_k$ and $\widehat{y}_k \leftarrow \nabla_p\mathcal{L}(p_{k+1}, \lambda_{k+1} - \nabla_p\mathcal{L}(p_k, \lambda_{k+1}))$, and obtain $B_{k+1}$ by updating $B_k$ using a quasi-Newton formula;
            $B_{k+1} = B_k + \frac{(\widehat{y}_k - B_k s_k)(\widehat{y}_k - B_k s_s)^T}{(\widehat{y}_k - B_k s_s)^T s_s}$;
18:     **end if**
    **end while**
```

1) Sensitivity Calculations: The first place, where appropriate using of parallel computing can improve time performance is connected with calculations of sensitivites. The important question is always what and why should be calculated. Firstly, SQP algorithm needs the gradients of both objective and constraints functions.

Estimation of the objective function and constraint functions can be obtained by using finite difference derivative approximation.

Consider a subroutine, that estimates the gradient of the objective function and constraint functions. This calculation involves computing function values at points near the current location of p. As a result there was obtained the vector

$$\nabla g(p) = \begin{bmatrix} \frac{g(p+\Delta_1 e_1) - g(p)}{\Delta_1} \\ \frac{g(p)+\Delta_2 e_2 - g(p)}{\Delta_2} \\ \vdots \\ \frac{g(p)+\Delta_n e_n - g(p)}{\Delta_n} \end{bmatrix}, \tag{18}$$

where

g	is objective or constraint function,
e_i	are the unit direction vectors,
Δ_i	is the size of a step in the e_i direction.

Gradient estimated by central differences can be obtained in the similar way

$$\nabla g(p) = \begin{bmatrix} \frac{g(x+\Delta_1 e_1) - g(x-\Delta_1 e_1)}{2\Delta_1} \\ \frac{g(x+\Delta_2 e_2) - g(x-\Delta_2 e_2)}{2\Delta_2} \\ \vdots \\ \frac{g(x+\Delta_n e_n) - g(x-\Delta_n e_n)}{2\Delta_n} \end{bmatrix}, \tag{19}$$

In this situation there is a possibility to compute successive elements of the gradient vector parallel. It is worthy to note, that these elements are independent and computing of every elements needs objective or constraint function evaluation.

Let us say, that the considered model is very complicated and calculation of every objective function is time-consuming. So, the question is, if it is better to use parallel computing for gradient estimation or parallel evaluation of objective function? The last question is, how to avoid complex inequality constraints. The answer is sought in the next section and in simulations of Catalyst Mixing Problem.

2) Multiple Shooting and Penalty function: In this section the Multiple shooting method (fig. 2) and its applications for solving both ordinary differential and differential-algebraic systems is considered.

The idea is to break the time domain $[t_I, t_F]$ into smaller intervals of the form

$$t_I = t_1 < t_2 < \cdots < t_{N_T} = t_F. \tag{20}$$

Let us denote y_0^l for $l = 1, \dots, N_T$ as the initial value for the dynamic variables at the beginning of every of the segment l. For segment l the DAE system can be solved from t_l to the end of the segment at t_{l+1}. The results of this integration can be denoted by $\widehat{y}_l$. Collecting all segments, let us define a set of NLP variables

$$p^T = (y_0^2, y_0^3, \dots, y_0^{N_T}). \tag{21}$$

The segments have to be joined at the boundaries, so the constraints are imposed

$$c(p) = \begin{bmatrix} y_0^2 - \widehat{y}_1 \\ y_0^3 - \widehat{y}_2 \\ \vdots \\ y_0^{N_T} - \widehat{y}_{N_T-1} \end{bmatrix}, \tag{22}$$

One of the results of the multiple shooting approach is an increase in the size of the NLP problem. There are additional variables and constraints, which were introduced for each segment. The number of NLP variables and constraints for a multiple shooting application is $n = n_y(N_T - 1)$, where n_y is the number of dynamic variables y and $(N_T - 1)$ is the number of segments. It is importanat to note, that the Jacobian matrix, which is needed to compute the Newton search direction, is sparse. Only $(N_T - 1)n_y^2$ elements in this matrix are nonzero of a possible $[(N_T - 1)n_y]^2$. Thus, the percentage of nonzeros is proportional to $\frac{1}{N_T - 1}$ indicating that the matrix get sparser as the number of intervals grows [4].

The immediate benefit of the multiple shooting algorithm is the ability to exploit a parallel processor. The method is sometimes called parallel shooting because the simulation of each element can be implemented on an individual processor [4].

The method presented in [4] was often used in other applications like in [12], [13]. Now we propose another approach to handle contraints, which are caused by multiple shooting method. The idea was previously introduced as one of the approaches for Mathematical Programming with Complementarity Constraints and results in augmented objective function.

In this approach the objective function was augmented by scalar product with penalty parameter.

$PF(\rho)$

$$\widehat{\phi}(p) = \phi(p) + \rho a^T b \rightarrow min, \tag{23}$$

$$c_i(p) = 0, i \in E, \tag{24}$$

$$c_i(p) \leq 0, i \in I. \tag{25}$$

$$a, b \geq 0, \tag{26}$$

where parameter ρ depends on the considered problem and $a^T b$ reflects the complementarity conditions.

In this manner the augmented objective function was constructed

$$\widehat{\phi}(p) = \phi(p) + \rho \sum_{l=1}^{N_T - 1} (y_0^{l+1} - \widehat{y}_l)^2 \rightarrow min, \tag{27}$$

$$c_i(p) = 0, i \in E, \tag{28}$$

$$c_i(p) \leq 0, i \in I. \tag{29}$$

In both approaches complex constraints were introduced to objective functionand enhanced by penalty parameter ρ, which is constant in all iterations.

B. DAE solver

The range of applications of this method depends on the applied DAE solver. The multiple shooting method allows to use parallel solving systems, hence the calculation of the objective function can be faster, especially when the function is complicated. Such approach may result in labor-intensive task due to the need of splitting the tasks in parallel and send to different processors and receive results. These activities occur during each time calculating the objective function.

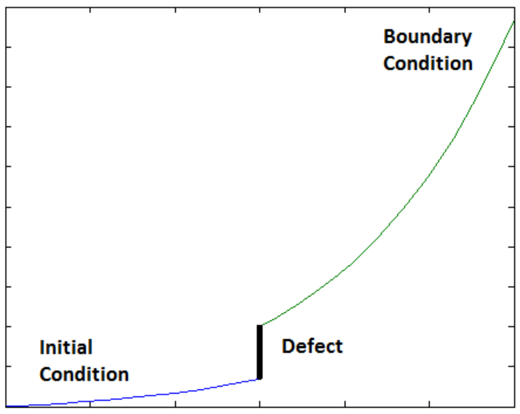

Figure 2. Multiple shooting method.

IV. CATALYST MIXING PROBLEM

This problem determines the optimal mixing policy of two catalysts along the length of a tubular reactor. The mixing ratio of the catalysts is the control variable [11].

This example used here does not contain any path constraint for state variables. The numerical algorithm of solving this problem reduces to the control parameterization method. Since there are no state variables involved in path constraint in the formulation, the discretization of the control variable and start points for multiple shooting methods leads to an NLP with upper and lower bounds on both types of decision variables [11].

The optimal mixing policy involves two switches. In the control parameterization approach, the controls can be restricted to piecewise constant, converting it to a three-variables problem [3]. In our approach, we allow more flexibility in the determination of the control trajectory and hence the simulations were performed for different numbers of control variables and shooting intervals.

The solution is sought for a fixed value of the reactor length. As the length increases, the structure of the solution can change drastically. For the greater length, the optimal control profile includes not only the portions at bounds but also a singular arc section in the middle . In an optimal control context, the singular arc exists whenever the Hamiltonian of the system does not depend on the control explicitly. Equivalently, the problem can also be regarded as having a high index portion because higher order derivatives of the state variables have to be evaluated to determine the decision variables . For this particular instance, the DAE solution contains two index one sections and one index three section between them [19].

The dynamic optimization formulation is described as follows:

$$\phi(p) = y_3(1.0) \rightarrow max, \tag{30}$$

subject to

$$\begin{aligned} \frac{dy_1}{dt} &= u(10y_2 - y_1), \\ \frac{dy_2}{dt} &= u(y_1 - 10y_2) - (1-u)y_2, \\ y_3 &= 1 - y_1 - y_2, \\ y(0) &= [1.0, 0.0, 0.0], \\ u(t) &\in [0.0, 1.0]. \end{aligned} \tag{31}$$

System (30)-(31) can be prescribed as ODE system

$$\phi(p) = 1 - y_1(1.0) - y_2(1.0) \rightarrow max, \tag{32}$$

subject to

$$\begin{aligned} \frac{dy_1}{dt} &= u(10y_2 - y_1), \\ \frac{dy_2}{dt} &= u(y_1 - 10y_2) - (1-u)y_2, \\ y(0) &= [1.0, 0.0], \\ u(t) &\in [0.0, 1.0]. \end{aligned} \tag{33}$$

What does the parameter p mean here? It is a vector of all variables. For Simultaneous Approach with 5 control functions, the vector p was steated as follows

$$p = [y_{01}^1, y_{02}^1, \ldots, y_{01}^5, y_{02}^5, u_1, \ldots, u_5]^T. \tag{34}$$

Simulations were performed on a processor Intel(R) Core(TM) i5 CPU 2.67 GHz in Matlab 7.11.0 with the following parameters

NLP	'TolFun',1e-7, 'TolX', 1e-7, 'Algorithm', 'sqp', 'Hessian', 'bfgs',
DAE	'RelTol',1e-7, 'AbsTol', 1e-7, 'BDF', 'on', 'MaxOrder', 1,
ρ	10^4.

The same task was solved with different numbers of continuous sections of the control function. Use of the multiple shooting method resulted in a significant number of variables to be considered. It is therefore proposed approach using parallel computing.

As can be seen, for small problems parallel computing of gradient of the objective function causes a significant increase in computation time. This is due to effort, which runs parallel processes. For the tasks with a large number of variables using parallel computing of the gradient with four parallel processes resulted in saving of computation time by half.

Details on the duration of the calculations are presented in the table I. The table II provides details about the course of the simulation, such as the number of iterations $iter$, the number of calculation of augmented objective function $feval$ and final value of differential state trajectories y_1 and y_2.

There are differential state trajectory for 100 piecewise constant control functions on the fig. 3

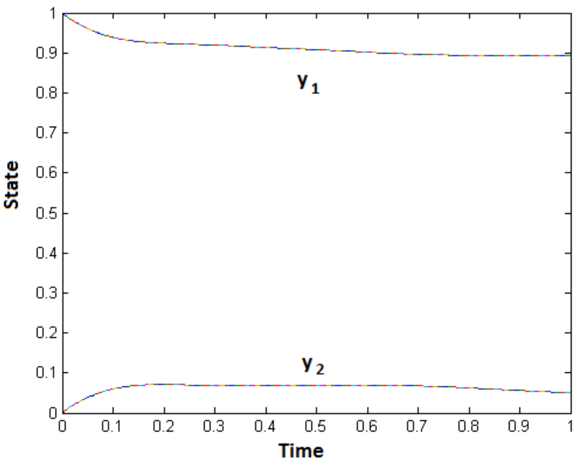

Figure 3. State trajectories. Case with 100 control functions.

Table I
PERFORMANCE TIME FOR PARALLEL COMPUTED GRADIENT

Number of control variables	without parallel processing [s]	2 parallel processors [s]	4 parallel processors [s]
1	5.1583	15.6728	15.7202
5	1.1254×10^3	717.9797	598.8218
25	6.2670×10^3	3.7672×10^3	2.9486×10^3
50	1.7984×10^4	1.0213×10^4	8.1255×10^3
100	7.6393×10^4	4.4441×10^4	3.4422×10^4

Table II
DETAILS OF COMPUTATIONS

Number of control variables	Number of variables	iter	feval	y_1	y_2
1	1	8	18	0.8883	0.0725
5	13	229	3484	0.8997	0.0524
25	73	274	21412	0.8984	0.0517
50	148	323	50410	0.8966	0.0514
100	298	472	145566	0.8934	0.0505

Trajectories of the control functions, which are constant in each time interval and discontinuous in a finite number of points, can be seen on the fig. 4-IV.

The studies have shown that starting and ending parallel processes in Matlab requires about 7.5 s for 2 processors and 9.5 s for 4 processors. For this reason, avoid frequent started parallel processes.

If for the gradient of the objective function parallel process is ran only once, while each time the calculation of the objective function in parallel requires initialization and termination of parallel computing. In the second case one can not be determined in advance how many times the objective function will be calculated.

The time, in which dynamic optimizatation problem can be solved, highly depends on used DAE and ODE solvers. In presented examples, with 100 control functions, the time not connected with solving DAE system, after solving DAE system to obtain objective function, is about $0.0003s$ and can be seen as relatively small. But this time depends lin-

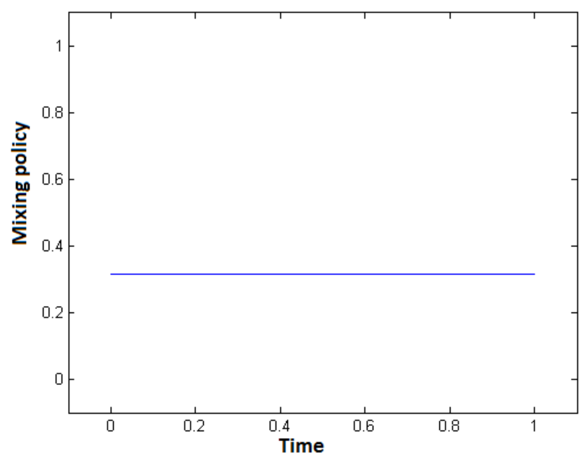

Figure 4. Control trajectory. Case with 1 control function.

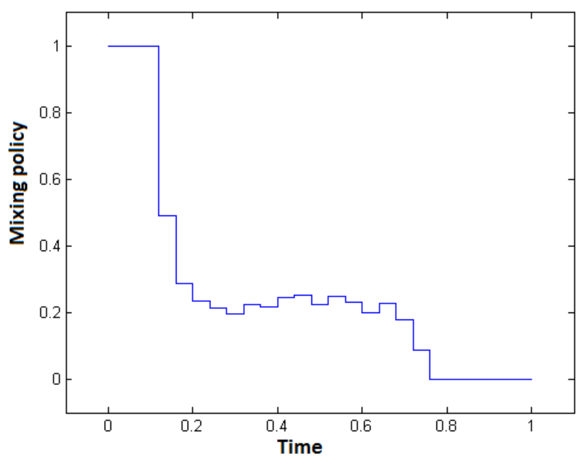

Figure 6. Control trajectory. Case with 25 control functions.

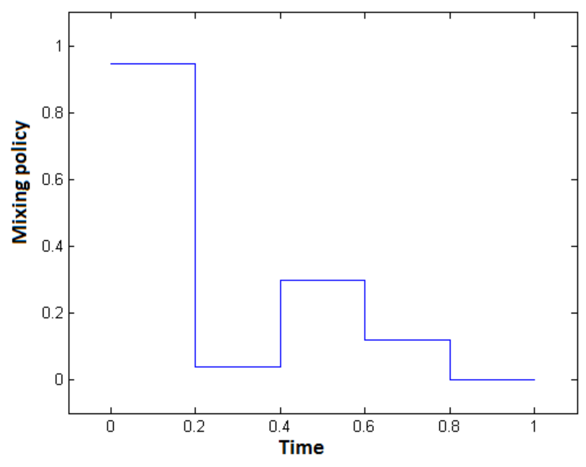

Figure 5. Control trajectory. Case with 5 control functions.

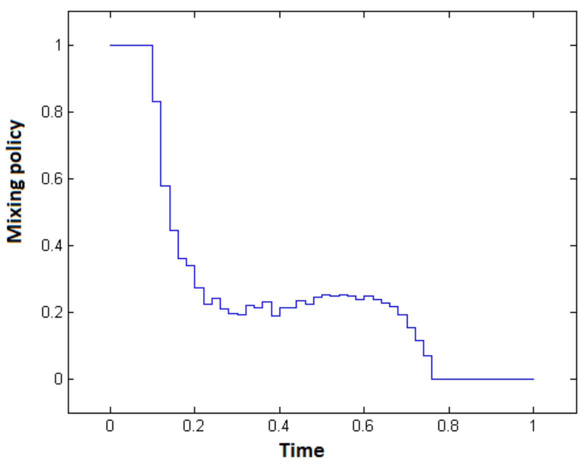

Figure 7. Control trajectory. Case with 50 control functions.

early on number of objective function evaluations and can not be minimized by parallel calculation of only the DAE system.

V. Conclusion

In the article considerations about parallel processing in solving optimal cotrol problems were made. There were discussed two options for parallelization of computing. The first is the ability to calculate the components of the gradient of the objective function parallel, the second was sought only in the parallel calculation of the DAE system. Simulations performed on an object derived from the chemical engineering have shown that the parallel calculation of the gradient of the objective function yields better results. It is caused by the amount of tasks sent to processors and received results. Equality constraints, arising from the use of the multiple shooting method, were introduced to the objective function.

VI. Acknowledgement

The authors are grateful to reviewers for constructive comments and suggestions.

One of the authors would like to thank Karol Maślanka for fruitful discussions about the issues described in this article.

References

[1] B.T. Baumrucker, J.G. Renfro, L.T. Biegler, "MPEC problem formulations and solution strategies with chemical engineering applications", *Computers and Chemical Engineering*, Vol. 32, 2008, pp. 2903-2913.
[2] B.T. Baumrucker, L.T. Biegler, "MPEC strategies for optimization of a class of hybrid dynamic systems", *Journal of Process Control*, Vol. 19,2009, pp. 1248-1256.
[3] M.L. Bell, R.W.H. Sargent, "Optimal control of inequality constrained DAE systems", *Computers and Chemical Engineering*, Vol. 24, 2000, pp. 2385-2404.
[4] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Second edition*, SIAM, Philadelphia 2010.
[5] L. T. Biegler, *Nonlinear Programming. Concepts, Algorithms, and Applications to Chemical Processes*, SIAM, Philadelphia 2010.

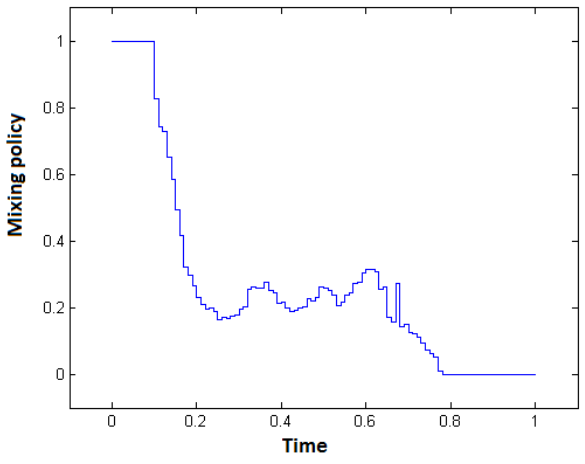

Figure 8. Control trajectory. Case with 100 control functions.

[6] K. E. Brenan, S. L. Campbell, L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia 1996.

[7] R. Fletcher, S. Leyffer, "Nonlinear programming without a penalty function", *Mathematical Programming*, Vol. 91, 2002, pp. 239-269.

[8] fmincon SQP Algorithm, MATLAB User's Guide, *MathWorks*, 2012.

[9] E. Hairer, C. Lubich, M. Roche, "The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods", *Lecture Notes in Mathematics*, Springer-Verlag, Berlin 1989.

[10] A. Hartwich, K. Stockmann, C. Terboven, S. Feuerriegel, W. Marquardt, "Parallel sensitivity analysis for efficient large-scale dynamic optimization", *Optimization and Engineering*, Vol. 12, 2011, pp. 489-508.

[11] Y. J. Huang , G.V. Reklaitis, , V. Venkatasubramanian, "Model decomposition based method for solving general dynamic optimization problems", *Computers and Chemical Engineering*, Vol. 26, 2002, pp. 863-873.

[12] D. B. Leineweber, I. Bauer, H. G. Bock, J. P. Schlöder, "An effcient multiple shooting based reduced SQP strategy for large scale dynamic process optimization. Part 1: theoretical aspects", *Computers and Chemical Engineering*, Vol. 27, 2003, pp. 157-166.

[13] D. B. Leineweber, A. Schäfer, H. G. Bock, J. P. Schlöder, "An effcient multiple shooting based reduced SQP strategy for large scale dynamic process optimization. Part II: Software aspects and applications", *Computers and Chemical Engineering*, Vol. 27, 2003, pp. 167-174.

[14] R. März, "Numerical methods for differential algebraic equations", *Acta Numerica*, 1992, pp. 141-198.

[15] J. Nocedal, S. J. Wright *Numerical Optimization. Second Edition,* Springer Verlag, 2006.

[16] L. Petzold, "Differential/Algebraic Equations are not ODEŠs", *SIAM J. Sci. and Stat. Comput.*, Vol. 3, 1982, pp. 367-384.

[17] A. U. Raghunathana, M. S. Diaz, L. T. Biegler, "An MPEC formulation for dynamic optimization of distillation operations", *Computers and Chemical Engineering*, Vol. 28, 2004, pp. 2037-2052.

[18] K. Styczeń, P. Drąg, "A modified multipoint shooting feasible-SQP method for optimal control of DAE systems", *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2011 pp. 477-484.

[19] P. Tanartkit, L. T. Biegler, "A nested, simultaneous approach for dynamic optimization problems - II : the outer problem", *Computers chem. Engng*, Vol. 21, 1997, pp. 1365-1388.

[20] V. S. Vassiliadis, R. W. H. Sargent, C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints", *Ind. Eng. Chem. Res.*, Vol. 33, No. 9, 1994, pp. 2111-2122.

[21] V. S. Vassiliadis, R. W. H. Sargent, C. C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems with Path Constraints", *Ind. Eng. Chem. Res.*, vol. 33, No. 9, 1994, pp. 2123-2133.