

Article

A Direct Optimization Algorithm for Problems with Differential-Algebraic Constraints: Application to Heat and Mass Transfer

Paweł Drąg 

Department of Control Systems and Mechatronics, Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland; pawel.drag@pwr.edu.pl

Received: 30 November 2020; Accepted: 14 December 2020; Published: 17 December 2020



Abstract: In this article, an optimization task with nonlinear differential-algebraic equations (DAEs) is considered. As a main result, a new solution procedure is designed. The computational procedure represents the sequential optimization approach. The proposed algorithm is based on a multiple shooting parametrization method. Two main aspects of a generalized parametrization approach are analyzed in detail: a control function and DAE model parametrization. A comparison between the original and modified DAEs is made. The new algorithm is applied to solve an optimization task in heat and mass transfer engineering.

Keywords: numerical optimization; differential-algebraic equations; parametrized DAE constraints; multiple shooting method; heat and mass transfer

1. Introduction

Currently, the dynamic development of engineering systems of parametric models can be observed. The parametric models, built from variables with a known interpretation connected with the equations of the laws of physics, are able to more and more accurately reflect the studied dependencies. The development of this type of model is limited only by the scope of natural science and computing capabilities. The parallel development of the mathematical models and numerical simulation algorithms can be easily observed in many fields of engineering, in particular in chemical engineering [1] and mechanical engineering [2]. Advanced computational techniques for the process design and optimization can have a positive impact on their quality and profitability.

Literature studies indicate the development of the parametric mathematical models of heat and mass transfer phenomena, which can be applied to HVAC (heat, ventilation, and air-conditioning) systems. The design research is oriented toward the systems operating under specific environmental conditions, as well as the simulation analysis of the system performance at various operating points in order to increase the efficiency of heat and mass exchange units. Therefore, the latest achievements in the use of the mass and heat transfer models are presented.

In their work, Pandelidis et al. [3] numerically analyzed the M-cycle cooling tower operation. In this research, critical heat and mass transfer process conditions were considered. Moreover, some critical factors with the highest impact on the system performance were established. Finally, a higher than 100% wet-bulb effectiveness for water cooling was achieved.

Yang and co-workers [4] designed some theoretical models of buoyancy-driven flows by area heat sources. According to the analyzed dependencies, a new law of thermal stratified flow for the transition zone was obtained. The numerical simulation approach was used to verify the correctness of the approximate theoretical solutions.

In [5], a highly specialized liquid desiccant air-conditioning system for bus spray-paint booths was considered. The adoption of the liquid desiccant dehumidifier can reduce the cooling load by 62.1% and be helpful in avoiding reheating.

El Loubani and co-workers [6] presented and analyzed a feasible hybrid cooling system for application to office spaces in hot and humid climates. The designed approach combined a phase change material thermal storage layer with a melting temperature of 25 °C to cool the supply air, as well as a personalized evaporative cooler to provide cooling for the occupants. Then, the designed solution was sized, implemented, and tested using the numerical simulation approach. As one of the main results, the designed solution achieved 87% energy savings compared to the conventional HVAC systems.

Schreiber et al. [7] applied reinforcement learning algorithms for load shifting in a cooling supply system. The work concluded that the obtained convergence speed is not suitable for complex planning learning of thermal energy networks in real time. Nonetheless, some standardized interfaces can facilitate the systematic testing of new computational tools for real-life energy systems.

In [8], a highly specialized case study was considered. A detailed computational model was used for the prediction of domestic hot water (DHW) heat. The prediction accuracy was improved by the additional artificial variables introduced. The different prediction techniques can be used to select the best DHW heat model.

In the mentioned research, the obtained mathematical parametric models consisted of the nonlinear differential-algebraic relations (differential-algebraic equations (DAEs)). Some special aspects related to the highly nonlinear DAE systems, like their numerical solution and optimization, are the subject of both theoretical and simulation research. The latest scientific achievements in the numerical simulation of the DAE systems have been reported in the literature.

A new general framework for dynamic optimization subject to nonlinear systems was designed by Kelley et al. [9]. To solve the considered task, a multiple shooting “backbone” strategy with discrete subproblems was proposed. Then, a Lagrangian relaxation (LR)-based decomposition scheme was analyzed. It was indicated that the proposed algorithm with the Lagrangian relaxation scheme makes the computations faster.

Modelica, as an object-oriented modeling language, can be used as a computational environment for the differential-algebraic models’ simulation [10]. Recently, Ploch et al. [11] considered the differential-algebraic equations with embedded optimization criteria as a special class of DAEs. In the presented situation, the systems with less algebraic equations than algebraic variables were analyzed. The authors proposed a new solution procedure, which was based on substituting the embedded optimization problem to obtain a nonsmooth DAE system. Finally, the nonsmooth system can be solved by a root-finding DAE solver.

Caspari et al. [2] considered the dynamic optimization problems with state-dependent complementarity constraints. It was indicated that the complementarity constraints can be formulated as smoothed nonlinear complementarity problem (NCP) functions. This approach resulted in smooth DAEs solved by the direct shooting-based algorithms. Moreover, the well-posedness of both nonsmooth and smoothed differential-algebraic equations was analyzed.

Optimization with a heat and mass transfer mathematical model is a difficult and actual research problem. Therefore, this work is aimed at designing a new solution procedure, which can address tasks with nonlinear differential-algebraic constraints. Some detailed mathematical derivations related to the nonlinear optimization task formulation are presented. The considerations try to take the reader from the general DAE constrained optimization task to the nonlinear optimization problem, which can be implemented using the currently available numerical optimization procedures. In other words, the appropriate objective function formulation with all accompanying constraints needs to be clearly defined. Then, in Section 2, the main features of the optimization task with DAE constraints are summarized. A generalized parametrization approach is introduced in Section 3. In Section 4, a modified optimization problem is formulated. Then, the new solution algorithm is designed

in Section 5. The results of numerical simulations are presented in Section 6. Finally, the conclusions are given in Section 7.

2. Optimization with Differential-Algebraic Constraints

The starting point for further considerations is the statement of a dynamic optimization task with the differential-algebraic constraints. In some practical applications, the DAE system can be presented in a semi-explicit form:

$$\begin{aligned}\dot{y}(t) &= f(y(t), z(t), u(t), p, t) \\ 0 &= g(y(t), z(t), u(t), p, t)\end{aligned}\quad (1)$$

where the specified functions and variables have appropriate properties and perform the assumed roles in the model; in particular, $y(t) \in \mathcal{R}^{n_y}$ is a vector of differential state trajectories, and $\dot{y}(t) = \frac{dy}{dt}$. Moreover, $z(t) \in \mathcal{R}^{n_z}$ is a vector of algebraic state trajectories. $u(t) \in \mathcal{R}^{n_u}$ denotes the vector of control functions. $p \in \mathcal{R}^{n_p}$ is a vector of parameters constant in time. $t \in \mathcal{R}$ is an independent variable with $t \in [t_0 \ t_f]$. The important assumptions are connected with the functions f and g and the vector of consistent initial conditions $[y(t_0) \ z(t_0)]^T = [y_0 \ z_0]^T$.

Assumption 1. The initial conditions $[y(t_0) \ z(t_0)]^T = [y_0 \ z_0]^T$ for the system (1) are consistent, iff for a given initial value of the control function $u(t_0)$, the equality:

$$g(y_0, z_0, u(t_0), p, t_0) = 0 \quad (2)$$

is fulfilled.

The consistent initialization of the differential-algebraic systems is an important and classical task considered in the literature. The vector $[y(t_0) \ z(t_0)]^T$ can be obtained as a result of analytical considerations [12], characterized by the real-life process initial conditions [13], computed by the deterministic optimization procedure [14] or obtained by heuristic optimization algorithms [15].

The relation between the initial conditions' consistency and the index of the considered system can be obtained.

Assumption 2. Let us assume that the following C^2 class functions f and g are considered:

$$\begin{aligned}f: \mathcal{R}^{n_y} \times \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_p} \times \mathcal{R} &\rightarrow \mathcal{R}^{n_y} \\ g: \mathcal{R}^{n_y} \times \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_p} \times \mathcal{R} &\rightarrow \mathcal{R}^{n_z}\end{aligned}\quad (3)$$

Theorem 1. Iff a matrix $\frac{\partial g}{\partial z}$ is nonsingular, then the system (1) is an implicit ODE of the following form:

$$\begin{aligned}\dot{y}(t) &= f(y(t), z(t), u(t), p, t) \\ \dot{z}(t) &= \left(\frac{\partial g}{\partial z}\right)^{-1} \left(-\frac{\partial g}{\partial y} \dot{y} - \frac{\partial g}{\partial u} \dot{u} - \frac{\partial g}{\partial t}\right)\end{aligned}\quad (4)$$

and has index-1.

Proof of Theorem 1. Let us differentiate the algebraic part of Equation (1) with respect to t :

$$0 = \frac{d}{dt} (g(y(t), z(t), u(t), p, t)) \quad (5)$$

which results in:

$$0 = \frac{\partial g}{\partial y} \dot{y} + \frac{\partial g}{\partial z} \dot{z} + \frac{\partial g}{\partial u} \dot{u} + \frac{\partial g}{\partial t} \quad (6)$$

and:

$$\dot{z}(t) = \left(\frac{\partial g}{\partial z} \right)^{-1} \left(-\frac{\partial g}{\partial y} \dot{y} - \frac{\partial g}{\partial u} \dot{u} - \frac{\partial g}{\partial t} \right) \quad (7)$$

This is true, iff a matrix $\frac{\partial g}{\partial z}$ is nonsingular. $\square$

Definition 1. The minimum number of times that all or part of the system (1) must be differentiated with respect to t in order to determine $\dot{z}(t)$ as a continuous function of z and t is the index of the DAE (1) [16].

From the computational methods' point of view, the consistency of the proposed initial conditions and the index of a system are the main properties of the differential-algebraic equations. The index-1 assumption is common for many real-life applications, although some index-2, as well as higher index formulations are the subject of research. Throughout this work, it is assumed that the considered model constraints can be described by the system of index-1 highly nonlinear DAEs in a semi-explicit form (1).

In the context of the differential-algebraic model constraints, a scalar-valued performance index in a general form can be considered:

$$\min_{u(t)} \mathcal{J}(y(t), z(t), u(t), t) \quad (8)$$

and in particular:

$$\mathcal{J}(y(t), z(t), u(t), t) = \phi(y(t_f), z(t_f), t_f) + \int_{t_0}^{t_f} \psi(y(t), z(t), u(t), t). \quad (9)$$

To optimize the performance index (8) subject to the DAE system (1), both indirect and direct methods can be applied. The indirect methods are based on the solution of the necessary optimality conditions. Thus, for the indirect methods, it is necessary to explicitly derive a system of the Euler–Lagrange equations consisting of adjoint equations, control equations, and transversality conditions, separately for each considered task [17]. Although the indirect methods have some known disadvantages, the recent research results indicate efficient indirect schemes [18], as well as combinations of direct-indirect approaches [19].

The optimization of systems described by the nonlinear differential-algebraic constraints requires a reliable approach to obtain the state trajectories for the given consistent initial conditions. Therefore, the further considerations related to the computational optimization with the nonlinear DAE model are based on the multiple shooting approach. The application of the multiple shooting step needs to divide the independent variable domain into an assumed number N of subintervals:

$$t^i \in [t_0^i \quad t_f^i], \quad i = 1, \dots, N, \quad (10)$$

according to the following rule:

$$t_0 = t_0^1 < t_f^1 = t_0^2 < t_f^2 \quad \dots \quad t_0^N < t_f^N = t_f \quad (11)$$

and can be used to parametrize the dynamic optimization task. An appropriate optimization problem parametrization enables us to apply efficient nonlinear optimization procedures and obtain a suboptimal solution even for highly nonlinear and possibly unstable systems. There are three main dynamic optimization task approaches

1. Control vector parametrization (CVP): The parametrization is related only to the control function and results in a medium-scale nonlinear optimization task [20].
2. Direct shooting method: The parametrized control function, as well as the initial conditions of the state trajectories are treated as additional decision variables. The differential-algebraic equations

are solved for each subinterval independently. Therefore, an efficient DAE solver is required. Additionally, the equality constraints providing the state trajectory continuity need to be taken into account [21].

3. Direct transcription formulation: The values of the state trajectories are parametrized and treated as additional decision variables. To parametrize the state trajectories, the Runge–Kutta, Hermite–Simpson, or any other numerical scheme can be used. The direct transcription formulation results in a large-scale and sparse nonlinear optimization task. Moreover, only the final result can be considered as an applicable solution [17].

Recently, a new family of methods was constituted by a combination of the direct shooting method with a DAE model approximation. The multiple shooting approach enables us to consider other systems in each subinterval. Moreover, the appropriate approximation makes the highly nonlinear DAE models possible to solve. Then, the new model properties can be analyzed from a computational optimization point of view.

3. A Generalized Parametrization Approach

In the presented work, the multiple shooting approach is used to parametrize both the control function, as well as the differential-algebraic model of the system. Usually, in the direct optimization algorithms, a partial state trajectory parametrization is often used. Therefore, to unify parametrization issues, the general method for the model, state, and control function parametrization is defined. The presented discussion is an extension for the understanding of the direct shooting approach.

Definition 2. Let us denote by $\mathcal{P}_m \{f(x)\}$ the parametrization procedure, where $m \in \mathcal{N}_+$ and $f : \mathcal{R}^{n_x} \rightarrow \mathcal{R}^{n_f}$. $\mathcal{P}_m \{f(x)\}$ results in a new parametrized function $\tilde{f}(P; x)$, such that $\tilde{f} : \mathcal{R}^{n_x} \rightarrow \mathcal{R}^{n_f}$ and $P \in \mathcal{R}^{n_f \times m}$ is a matrix of the function $\tilde{f}$ parameters, where:

$$\min_P \int_{x_0}^{x_f} \|f(x) - \tilde{f}(P, x)\| \quad (12)$$

for $x \in [x_0 \ x_f]$ and according to the chosen norm. In particular, the parametrization of a scalar function $f(x)$ such that $f : \mathcal{R}^{n_x} \rightarrow \mathcal{R}$ is:

$$\mathcal{P}_m \{f(x)\} = \tilde{f}([p^1 \ p^2 \ \dots \ p^m]; x) = \tilde{f}(P; x). \quad (13)$$

Then, according to Definition 2, the control function $u(t) \in \mathcal{R}^{n_u}$ can take a new form:

$$\begin{aligned} \mathcal{P}_{m_u} \{u(t)\} &= \mathcal{P}_{m_u} \left\{ \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_{n_u}(t) \end{bmatrix} \right\} = \begin{bmatrix} \mathcal{P}_{m_u} \{u_1(t)\} \\ \mathcal{P}_{m_u} \{u_2(t)\} \\ \vdots \\ \mathcal{P}_{m_u} \{u_{n_u}(t)\} \end{bmatrix} = \begin{bmatrix} \tilde{u}_1([u_1^1 \ u_1^2 \ \dots \ u_1^{m_u}]; t) \\ \tilde{u}_2([u_2^1 \ u_2^2 \ \dots \ u_2^{m_u}]; t) \\ \vdots \\ \tilde{u}_{n_u}([u_{n_u}^1 \ u_{n_u}^2 \ \dots \ u_{n_u}^{m_u}]; t) \end{bmatrix} \\ &= \tilde{u} \left(\underbrace{\begin{bmatrix} u_1^1 & u_1^2 & \dots & u_1^{m_u} \\ u_2^1 & u_2^2 & \dots & u_2^{m_u} \\ \vdots & \vdots & \dots & \vdots \\ u_{n_u}^1 & u_{n_u}^2 & \dots & u_{n_u}^{m_u} \end{bmatrix}}_{P_u \in \mathcal{R}^{n_u \times m_u}}; t \right) = \tilde{u}(P_u; t) \end{aligned} \quad (14)$$

and briefly:

$$\mathcal{P}_{m_u} \{u(t)\} = \tilde{u}(P_u; t). \quad (15)$$

The parameter matrix P_u should be found with the condition:

$$\min_{P_u} \int_{t_0}^{t_f} \|u(t) - \tilde{u}(P_u; t)\| \quad (16)$$

for $t \in [t_0 \quad t_f]$ and according to the chosen norm.

Moreover, the parametrization procedure $\mathcal{P}$ can be applied to the DAE constraints (1). In other words, it can be used with regard to the model relations, not the state trajectories. The parametrization procedure $\mathcal{P}$ can result in a new form of the model dependencies:

$$\begin{aligned} \mathcal{P}_m \left\{ \begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} \right\} &= \mathcal{P}_m \left\{ \begin{bmatrix} f_1(y(t), z(t), u(t), p, t) \\ f_2(y(t), z(t), u(t), p, t) \\ \vdots \\ f_{n_y}(y(t), z(t), u(t), p, t) \\ g_1(y(t), z(t), u(t), p, t) \\ g_2(y(t), z(t), u(t), p, t) \\ \vdots \\ g_{n_z}(y(t), z(t), u(t), p, t) \end{bmatrix} \right\} = \begin{bmatrix} \mathcal{P}_m \{f_1(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_m \{f_2(y(t), z(t), u(t), p, t)\} \\ \vdots \\ \mathcal{P}_m \{f_{n_y}(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_m \{g_1(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_m \{g_2(y(t), z(t), u(t), p, t)\} \\ \vdots \\ \mathcal{P}_m \{g_{n_z}(y(t), z(t), u(t), p, t)\} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{f}_1(P_M; y(t), z(t), u(t), p, t) \\ \tilde{f}_2(P_M; y(t), z(t), u(t), p, t) \\ \vdots \\ \tilde{f}_{n_y}(P_M; y(t), z(t), u(t), p, t) \\ \tilde{g}_1(P_M; y(t), z(t), u(t), p, t) \\ \tilde{g}_2(P_M; y(t), z(t), u(t), p, t) \\ \vdots \\ \tilde{g}_{n_z}(P_M; y(t), z(t), u(t), p, t) \end{bmatrix}, \end{aligned} \quad (17)$$

where:

$$P_M = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^m \\ a_2^1 & a_2^2 & \cdots & a_2^m \\ \vdots & \vdots & \cdots & \vdots \\ a_{n_y}^1 & a_{n_y}^2 & \cdots & a_{n_y}^m \\ a_{n_y+1}^1 & a_{n_y+1}^2 & \cdots & a_{n_y+1}^m \\ a_{n_y+2}^1 & a_{n_y+2}^2 & \cdots & a_{n_y+2}^m \\ \vdots & \vdots & \cdots & \vdots \\ a_{n_y+n_z}^1 & a_{n_y+n_z}^2 & \cdots & a_{n_y+n_z}^m \end{bmatrix} \in \mathcal{R}^{(n_y+n_z) \times m}. \quad (18)$$

Finally, in the considered DAE model (17), the parametrized control function $\tilde{u}(P_u; t)$ can be introduced:

$$\begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{f}(P_M; y(t), z(t), \mathcal{P}_{m_u}\{u(t)\}, p, t) \\ \tilde{g}(P_M; y(t), z(t), \mathcal{P}_{m_u}\{u(t)\}, p, t) \end{bmatrix} = \begin{bmatrix} \tilde{f}(P_M; y(t), z(t), \tilde{u}(P_u; t), p, t) \\ \tilde{g}(P_M; y(t), z(t), \tilde{u}(P_u; t), p, t) \end{bmatrix}. \quad (19)$$

Let us consider the special case, when $\mathcal{P}_m = \mathcal{P}_{n_y+n_z+n_u}$. Then, the DAE model (1) takes a special parametrized form:

$$\begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{P}_{n_y+n_z+n_u} \{f_1(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_{n_y+n_z+n_u} \{f_2(y(t), z(t), u(t), p, t)\} \\ \vdots \\ \mathcal{P}_{n_y+n_z+n_u} \{f_{n_y}(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_{n_y+n_z+n_u} \{g_1(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_{n_y+n_z+n_u} \{g_2(y(t), z(t), u(t), p, t)\} \\ \vdots \\ \mathcal{P}_{n_y+n_z+n_u} \{g_{n_z}(y(t), z(t), u(t), p, t)\} \end{bmatrix} = \begin{bmatrix} \tilde{f}_1(P_{DAE}; y(t), z(t), u(t), p, t) \\ \tilde{f}_2(P_{DAE}; y(t), z(t), u(t), p, t) \\ \vdots \\ \tilde{f}_{n_y}(P_{DAE}; y(t), z(t), u(t), p, t) \\ \tilde{g}_1(P_{DAE}; y(t), z(t), u(t), p, t) \\ \tilde{g}_2(P_{DAE}; y(t), z(t), u(t), p, t) \\ \vdots \\ \tilde{g}_{n_z}(P_{DAE}; y(t), z(t), u(t), p, t) \end{bmatrix}, \quad (20)$$

where:

$$P_{DAE} = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^{n_y+n_z+n_u} \\ a_2^1 & a_2^2 & \cdots & a_2^{n_y+n_z+n_u} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n_y}^1 & a_{n_y}^2 & \cdots & a_{n_y}^{n_y+n_z+n_u} \\ a_{n_y+1}^1 & a_{n_y+1}^2 & \cdots & a_{n_y+1}^{n_y+n_z+n_u} \\ a_{n_y+2}^1 & a_{n_y+2}^2 & \cdots & a_{n_y+2}^{n_y+n_z+n_u} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n_y+n_z}^1 & a_{n_y+n_z}^2 & \cdots & a_{n_y+n_z}^{n_y+n_z+n_u} \end{bmatrix} \in \mathcal{R}^{(n_y+n_z) \times (n_y+n_z+n_u)} \quad (21)$$

with the condition:

$$\min_{P_{DAE}} \int_{t_0}^{t_f} \left\| \begin{bmatrix} f(y(t), z(t), u(t), p, t) \\ g(y(t), z(t), u(t), p, t) \end{bmatrix} - \begin{bmatrix} \tilde{f}(P_{DAE}; y(t), z(t), u(t), p, t) \\ \tilde{g}(P_{DAE}; y(t), z(t), u(t), p, t) \end{bmatrix} \right\|. \quad (22)$$

As was indicated, the parametrization $\mathcal{P}_{n_y+n_z+n_u}$ is applied to the model, not to the state trajectory. The appropriate size of the matrix P_{DAE} can be used to obtain a new, linear model of the DAE constraints:

$$\begin{aligned} \begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} &= P_{DAE} \begin{bmatrix} y(t) \\ z(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \begin{bmatrix} y(t) \\ z(t) \\ u(t) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}}_{A \in \mathcal{R}^{(n_y+n_z) \times (n_y+n_z)}} \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} + \underbrace{\begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix}}_{B \in \mathcal{R}^{(n_y+n_z) \times (n_u)}} \begin{bmatrix} u(t) \end{bmatrix}, \end{aligned} \quad (23)$$

and similar to the state space equations. The matrix P_{DAE} (21) manipulations can result in other forms of the considered dynamical system. Finally, the control vector parametrization $\mathcal{P}_{m_u}$ can be formally applied in the parametrized model (23)

$$\begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} = A \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} + B \mathcal{P}_{m_u} \{u(t)\} = A \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} + B \tilde{u}(P_u; t), \quad (24)$$

where A , B , and P_u represent matrices of a known size, but with unknown values of their components.

4. The New Computational Procedure

The presented, general approach to the differential-algebraic constraint parametrization is used to design a new computational procedure. The described algorithm is not problem-specialized. Therefore, the appropriate parametrization approach predestines it to be used in various areas of technology.

The formulation of the new solution procedure is based on three aspects of the presented considerations:

- the multiple shooting approach,
- the DAE constraint parametrization,
- the similarity between the original DAE and the parametrized system.

There are compatible mile-stones, representing appropriate steps in the new optimization task formulation. All presented aspects influence the final form of the modified optimization task, which should be characterized by a high stability and make a nonlinear dynamics system simulation possible. Therefore, it will be clearly indicated how the multiple shooting-based parametrization can be used to obtain the modified optimization problem with features similar to the original one.

4.1. The Multiple Shooting Approach

The multiple shooting approach MSM_N enables us to divide the independent variable domain into an assumed number of N subintervals. Then, the relation between obtained subintervals takes the form of Equations (10) and (11). Moreover, in a sense, the differential-system of Equation (1) is divided into subintervals also. In the other words, the DAE systems can be considered independently in subintervals. This situation is presented with details in Equation (25):

$$\begin{array}{lcl}
 \text{EQs:} & \dot{y}(t) & = f(y(t), z(t), u(t), p, t) \\
 & 0 & = g(y(t), z(t), u(t), p, t) \\
 \text{TI:} & t & \in [t_0 \quad t_f] \\
 \text{CIC:} & 0 & = g(y(t_0), z(t_0), u(t_0), p, t_0) \\
 \text{IC:} & y(t_0) & = y_0
 \end{array} \xrightarrow{MSM_N}
 \left\{ \begin{array}{l}
 i = 1 \\
 \text{EQs:} \quad \dot{y}^1(t^1) = f^1(y^1(t^1), z^1(t^1), u^1(t^1), p, t^1) \\
 \quad \quad \quad 0 = g^1(y^1(t^1), z^1(t^1), u^1(t^1), p, t^1) \\
 \text{TI:} \quad \quad t^1 \in [t_0^1 \quad t_f^1] \\
 \text{CIC:} \quad \quad 0 = g^1(x_{y^1}, z^1(t_0^1), u^1(t_0^1), p, t_0^1) \\
 \text{IC:} \quad \quad y^1(t_0^1) = x_{y^1} \\
 \vdots \\
 i = N \\
 \text{EQs:} \quad \dot{y}^N(t^N) = f^N(y^N(t^N), z^N(t^N), u^N(t^N), p, t^N) \\
 \quad \quad \quad 0 = g^N(y^N(t^N), z^N(t^N), u^N(t^N), p, t^N) \\
 \text{TI:} \quad \quad t^N \in [t_0^N \quad t_f^N] \\
 \text{CIC:} \quad \quad 0 = g^N(x_{y^N}, z^N(t_0^N), u^N(t_0^N), p, t_0^N) \\
 \text{IC:} \quad \quad y^N(t_0^N) = x_{y^N}
 \end{array} \right. \quad (25)$$

The new system obtained by the MSM_N approach consists of the following components:

- $i = 1, \dots, N$ is the number of the current subintervals.
- The current i -th subinterval is considered in a time interval (TI) $t^i \in [t_0^i \quad t_f^i]$.
- The initial conditions (ICs) of the differential state trajectories are treated as decision variables and take a structured concatenated form:

$$\begin{bmatrix} y^1(t_0^1) \\ \vdots \\ y^N(t_0^N) \end{bmatrix} = \begin{bmatrix} x_{y^1} \\ \vdots \\ x_{y^N} \end{bmatrix} \in \mathcal{R}^{(n_{y^1} + \dots + n_{y^N})} \quad (26)$$

- EQs denote the system of the DAE constraints connected to the i -th subinterval. The considered system consists of the vector-valued functions f^i and g^i such that:

$$\begin{aligned} f^i : \mathcal{R}^{n_{y^i}} \times \mathcal{R}^{n_{z^i}} \times \mathcal{R}^{n_{u^i}} \times \mathcal{R}^{n_p} \times \mathcal{R} &\rightarrow \mathcal{R}^{n_{y^i}} \\ g^i : \mathcal{R}^{n_{y^i}} \times \mathcal{R}^{n_{z^i}} \times \mathcal{R}^{n_{u^i}} \times \mathcal{R}^{n_p} \times \mathcal{R} &\rightarrow \mathcal{R}^{n_{z^i}} \end{aligned} \quad (27)$$

and $y^i(t^i) \in \mathcal{R}^{n_{y^i}}$ is the vector of the differential state trajectories in the i -th subinterval. $z^i(t^i) \in \mathcal{R}^{n_{z^i}}$ is the vector of the algebraic state trajectories in the i -th subinterval. $u^i(t^i) \in \mathcal{R}^{n_{u^i}}$ denotes the vector of the control functions in the i -th subinterval. $p \in \mathcal{R}^{n_p}$ is the vector of the parameters constant in time. $t^i \in \mathcal{R}$ is the independent variable, and $t^i \in [t_0^i \ t_f^i]$.

- It is assumed that:

$$f^1 \neq f^2 \neq \dots \neq f^N \quad (28)$$

$$g^1 \neq g^2 \neq \dots \neq g^N \quad (29)$$

$$y^1(t^1) \neq y^2(t^2) \neq \dots \neq y^N(t^N) \quad (30)$$

$$z^1(t^1) \neq z^2(t^2) \neq \dots \neq z^N(t^N) \quad (31)$$

$$u^1(t^1) \neq u^2(t^2) \neq \dots \neq u^N(t^N) \quad (32)$$

and this is true, e.g., for DAE models in multi-stage systems [22].

- To solve the DAE system on the appropriate subintervals (25), the consistent initial conditions (CICs) are needed. In particular, the consistent initial conditions can be obtained with the following equations:

$$\begin{bmatrix} g^1(x_{y^1}, z^1(t_0^1), u^1(t_0^1), p, t_0^1) \\ \vdots \\ g^N(x_{y^N}, z^N(t_0^N), u^N(t_0^N), p, t_0^N) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (33)$$

- The continuity conditions: The multiple shooting-based algorithm should result in continuous trajectories of the state and control variables, unless explicitly stated otherwise. This means that the value of the state trajectory at the end of the current subinterval is equal to the state trajectory at the beginning of the next subinterval. The indicated relation takes the form of an equality constraint vector:

$$\begin{bmatrix} y^1(t_f^1) - x_{y^2} \\ \vdots \\ y^{N-1}(t_f^{N-1}) - x_{y^N} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (34)$$

4.2. The DAE Constraints' Parametrization

The presented MSM_N -based step of the new optimization task formulation does not indicate the exact way of the state and control trajectories' parametrization. These aspects will be considered in this section.

The parametrization used should take into account the properties of the considered optimization task, in particular:

- the physical (technological) properties of the system under consideration (e.g., oscillations, discontinuities, a nonlinear behavior like hysteresis, regions of instability),
- the properties of the DAE solver used (the ability to determine the consistent initial conditions, the model solution accuracy in the instability region),

- the performance of the optimization algorithm (the available operating memory, the possibility of parallel computations, dedicated numerical algebra procedures for sparse matrices and large-scale tasks, the globalization strategy, the rate of convergence).

The solutions presented in the literature represent two possible ways to parametrize the differential-algebraic constraints system (1).

1. Parametrize both the state and control trajectories, but without parametrization of the relations represented by the f and g functions. This approach results in a large-scale algebraic system of equations.

Let us denote:

$$\begin{aligned}\mathcal{P}_{m_y}\{y(t)\} &= \tilde{y}(P_y; t) \\ \mathcal{P}_{m_z}\{z(t)\} &= \tilde{z}(P_z; t) \\ \mathcal{P}_{m_u}\{u(t)\} &= \tilde{u}(P_u; t)\end{aligned}\quad (35)$$

where $m_y, m_z, m_u \in \mathcal{N}_+$ and P_y, P_z, P_u are the matrices of the approximating polynomial coefficients. Then, if:

$$\dot{\tilde{y}}(P_y; t_i) = \left. \frac{d\tilde{y}(P_y; t)}{dt} \right|_{t=t_i} \quad (36)$$

$t \in \{t_i\}_{i=0}^m$, with the relation $t_0 < \dots < t_m$, where $m = \max\{m_y, m_z, m_u\}$, the following algebraic system of equations $\mathcal{G}(P_y, P_z, P_u)$ is obtained:

$$\mathcal{G}(P_y, P_z, P_u) = \begin{bmatrix} \tilde{y}(P_y; t_0) - f(\tilde{y}(P_y; t_0), \tilde{z}(P_z; t_0), \tilde{u}(P_u; t_0), p, t_0) \\ \vdots \\ \tilde{y}(P_y; t_m) - f(\tilde{y}(P_y; t_m), \tilde{z}(P_z; t_m), \tilde{u}(P_u; t_m), p, t_m) \\ g(\tilde{y}(P_y; t_0), \tilde{z}(P_z; t_0), \tilde{u}(P_u; t_0), p, t_0) \\ \vdots \\ g(\tilde{y}(P_y; t_m), \tilde{z}(P_z; t_m), \tilde{u}(P_u; t_m), p, t_m) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (37)$$

2. The parametrization of the constraints represented by the functions f and g : Opposite the previous point, the state and control trajectories can be calculated by a numerical procedure implemented in a DAE solver. Recently, a special case of DAE model parametrization was used to introduce the variability constraints into the optimization task [22].

The choice of a particular parametrization method is dependent on the task under consideration, as well as the efficiency of the solution method used. The situation when the DAE constraints cannot be solved with the available implemented numerical procedure seems to be particularly interesting. Further considerations will be made to resolve this issue.

Let us consider the nonlinear system of differential-algebraic constraints (1). In computer simulations of many real-life problems, the direct application of the numerical DAE solvers cannot result in an accurate and stable solution. These cases were reported, e.g., in the monograph [22]. Therefore, it is suggested to apply a simplified DAE model, which can reflect the dynamical features appropriately and does not exhibit numerical difficulties or instabilities. The commonly used methodologies in the context of a highly nonlinear DAE constraint solution are based on a polynomial representation of the state trajectories $y(t)$ and $z(t)$ ([23]) or a full state parametrization (direct transcription method [17]). There is a common way to transform a difficult problem of solving the DAE system into a system of nonlinear algebraic relations. Then, the iterative methods can be used to improve the current solution step-by-step [24].

Opposite the presented procedures, the direct shooting approach (a combination of the multiple shooting method with the DAE solver) can be applied. It is assumed that the solution of the DAE system obtained on a shorter subinterval can be more accurate than a solution calculated for a long domain. However, even in this reduced situation, the DAE solver can fail and result in no solution, and the optimization procedure can be interrupted. Finally, the number of shooting points should be increased, and the whole solution procedure needs to be started again. Therefore, a combination of the multiple shooting method with the modified (parametrized) DAE model solved by an outer numerical procedure is considered.

In particular, the modification of the DAE system is not only understood as its linearization:

$$\begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} = A \Big|_{(y(t_\lambda), z(t_\lambda), u(t_\lambda))} \begin{bmatrix} y(t) \\ z(t) \end{bmatrix} + B \Big|_{(y(t_\lambda), z(t_\lambda), u(t_\lambda))} u(t), \quad (38)$$

where $t_\lambda \in [t_0 \ t_f]$. The linearization of the model constraints should be performed numerically or analytically at the beginning of the computations. Moreover, the obtained linearized model is dependent on the chosen simulation point $(y(t_\lambda), z(t_\lambda), u(t_\lambda))$.

One of the main ideas considered in this research is to estimate the unknown model parameters. This approach enables us to apply all forms of mathematical models, which can be effectively treated by the implemented outer DAE solver. The proposed parametrization approach results in the following theorem.

Theorem 2. *The parametrization of the DAE system in a semi-explicit form (1) with the $\mathcal{P}_{m_f}$ and $\mathcal{P}_{m_g}$ parametrizations for the differential and algebraic model constraints, respectively, introduces:*

$$n_y \times m_f + n_z \times m_g \quad (39)$$

decision variables, where $m_f, m_g \in \mathcal{N}_+$.

Proof of Theorem 2. Let us note that:

$$\begin{aligned} \begin{bmatrix} \dot{y}(t) \\ 0 \end{bmatrix} &= \begin{bmatrix} \mathcal{P}_{m_f} \{f(y(t), z(t), u(t), p, t)\} \\ \mathcal{P}_{m_g} \{g(y(t), z(t), u(t), p, t)\} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{f}(P_f; y(t), z(t), u(t), p, t) \\ \tilde{g}(P_g; y(t), z(t), u(t), p, t) \end{bmatrix}, \end{aligned} \quad (40)$$

where $m_f, m_g \in \mathcal{N}_+$ and $P_f \in \mathcal{R}^{n_y \times m_f}$, $P_g \in \mathcal{R}^{n_z \times m_g}$ are the matrices of unknown parameters. $\square$

4.3. The Similarity between the Original and Parametrized DAE Systems

An important issue is related to the method of measuring the similarity between the original DAE system and the modified model with the estimated coefficient matrices P_f and P_g . As was mentioned in the previous section, the difference between the systems can have an integral form adapted to the multiple shooting method, according to the following definition.

Definition 3. The difference between the original and modified DAE systems in a semi-explicit form (1) can be expressed as:

$$\Delta_{DAE} = \sum_{i=1}^N \sum_{j=1}^{n_y} \int_{t_0^i}^{t_f^i} \|f_j^i(\cdot) - \tilde{f}_j^i(\cdot)\| + \sum_{i=1}^N \sum_{j=1}^{n_z} \int_{t_0^i}^{t_f^i} \|g_j^i(\cdot) - \tilde{g}_j^i(\cdot)\|, \quad (41)$$

where $\|\cdot\|$ denotes a chosen norm, i is the index of a subinterval, and j denotes the index of the function vector component.

Further considerations require the following assumption.

Assumption 3. Functions $f_j^i(\cdot, t^i)$, $\tilde{f}_j^i(\cdot, t^i)$, $g_k^i(\cdot, t^i)$ and $\tilde{g}_k^i(\cdot, t^i)$ for $i = 1, \dots, N$, $j = 1, \dots, n_y$, $k = 1, \dots, n_z$ are continuous, real-valued, and defined on the appropriate subintervals $t^i \in [t_0^i, t_f^i]$ for all considered consistent initial conditions and control functions $u^i(t^i)$ and $\tilde{u}^i(\cdot, t^i)$.

Based on Assumption 3, let us consider the two following DAE constraint systems:

$$\begin{aligned} \dot{y}(t) &= f(y(t), z(t), \tilde{u}(P_u; t), p, t) \\ 0 &= g(y(t), z(t), \tilde{u}(P_u; t), p, t) \end{aligned} \quad (42)$$

and:

$$\begin{aligned} \dot{\tilde{y}}(t) &= \tilde{f}(P_f; \tilde{y}(t), \tilde{z}(t), \tilde{u}(P_u; t), p, t) \\ 0 &= \tilde{g}(P_g; \tilde{y}(t), \tilde{z}(t), \tilde{u}(P_u; t), p, t) \end{aligned} \quad (43)$$

To estimate the difference between the differential state trajectories:

$$\Delta_y = y - \tilde{y} \quad (44)$$

the implications of the mean value theorem can be applied.

Theorem 3. If $\dot{y}(t) = \dot{\tilde{y}}(t)$ for all t in an interval $t \in (t_0, t_f)$ of the domain of these functions, then $y(t) - \tilde{y}(t)$ is constant or $y(t) = \tilde{y}(t) + c$, where $c \in \mathcal{R}$ is constant on (t_0, t_f) .

Proof. Let $\Delta_y = y - \tilde{y}$, then $\Delta_{\dot{y}} = \dot{y} - \dot{\tilde{y}} = f(\cdot) - \tilde{f}(\cdot) = 0$ on the interval (t_0, t_f) . Therefore, $\Delta_y = y - \tilde{y} = f - \tilde{f} = \text{const}$ or $y = \tilde{y} + c$, constant $c \in \mathcal{R}$. $\square$

As a result of Theorem 3, to supervise the fulfillment of the condition:

$$\begin{bmatrix} \dot{y}^1(t^1) &= & \dot{\tilde{y}}^1(t^1) \\ \vdots & & \vdots \\ \dot{y}^N(t^N) &= & \dot{\tilde{y}}^N(t^N) \end{bmatrix} \quad (45)$$

the vector of equality constraints is considered as a part of the new optimization task:

$$\left[\begin{array}{l} f^1(y^1(t_0^1), z^1(t_0^1), \tilde{u}^1(P_u; t_0^1), p, t_0^1) - \tilde{f}^1(P_f; \tilde{y}^1(t_0^1), \tilde{z}^1(t_0^1), \tilde{u}^1(P_u; t_0^1), p, t_0^1) \\ f^1(y^1(t_f^1), z^1(t_f^1), \tilde{u}^1(P_u; t_f^1), p, t_f^1) - \tilde{f}^1(P_f; \tilde{y}^1(t_f^1), \tilde{z}^1(t_f^1), \tilde{u}^1(P_u; t_f^1), p, t_f^1) \\ \vdots \\ f^N(y^N(t_0^N), z^N(t_0^N), \tilde{u}^N(P_u; t_0^N), p, t_0^N) - \tilde{f}^N(P_f; \tilde{y}^N(t_0^N), \tilde{z}^N(t_0^N), \tilde{u}^N(P_u; t_0^N), p, t_0^N) \\ f^N(y^N(t_f^N), z^N(t_f^N), \tilde{u}^N(P_u; t_f^N), p, t_f^N) - \tilde{f}^N(P_f; \tilde{y}^N(t_f^N), \tilde{z}^N(t_f^N), \tilde{u}^N(P_u; t_f^N), p, t_f^N) \end{array} \right] \quad (46)$$

Remark 1. The values of $z^i(t_0^i)$, $i = 1, \dots, N$ are a consequence of the consistent initial conditions.

5. The New Solution Procedure

The main steps of the new solution algorithm are described briefly as “PRE $\xrightarrow{\text{Procedure}}$ POST”, where “PRE” denotes an input of the “Procedure”. Then, “POST” is the result of the “Procedure” for the input “PRE”.

The parametrized DAE-multiple shooting method algorithm:

1. Define the index-1 DAE constraints in the form (1).
2. For the system (1), find the values $n_y, n_z, n_u \in \mathcal{N}_+$.
3. Define the number $N \in \mathcal{N}_+$, which denotes the number of considered subintervals:

$$\begin{array}{ll} \text{EQs:} & \begin{array}{l} \dot{y}^i(t^i) = f^i(y^i(t^i), z^i(t^i), u^i(t^i), p, t^i) \\ 0 = g^i(y^i(t^i), z^i(t^i), u^i(t^i), p, t^i) \end{array} \\ \text{TI:} & t^i = [t_0^i \quad t_f^i] \\ \text{DAE(1), } N \xrightarrow{\text{MSM}_N} & \text{IC: } y^i(t_0^i) = x_{y^i} \\ & \text{CIC: } 0 = g^i(x_{y^i}, z^i(t_0^i), u^i(t_0^i), p, t_0^i) \\ & i = 1, \dots, N \end{array} \quad (47)$$

4. Implement the continuity constraints in the form (34).
5. Perform the control vector parametrization. Define $m_u \in \mathcal{N}_+$, then:

$$u(t), m_u \xrightarrow{\mathcal{P}_{m_u}} \tilde{u}(P_u, t), \quad (48)$$

where $P_u \in \mathcal{R}^{(n_y+n_z) \times m_u}$.

6. The DAE model parametrization: Define $m_f, m_g \in \mathcal{N}_+$, then:

$$\begin{array}{ll} f(\cdot), m_f & \xrightarrow{\mathcal{P}_{m_f}} \tilde{f}(P_f; \cdot) \\ g(\cdot), m_g & \xrightarrow{\mathcal{P}_{m_g}} \tilde{g}(P_g; \cdot) \end{array} \quad (49)$$

where $P_f \in \mathcal{R}^{n_y \times m_f}$ and $P_g \in \mathcal{R}^{n_z \times m_g}$.

7. The DAE models' similarity: Define the vector of the equality constraints according to Equation (46).
8. The objective function: The expressions presented in Equations (8) and (9) have a new form:

$$\min_{\substack{x_{y^i}, P_{f^i}, P_{g^i}, P_{u^i} \\ i = 1, \dots, N}} \tilde{\mathcal{J}}(y(t), z(t), \tilde{u}(P_u, t), t) \quad (50)$$

and, in particular:

$$\tilde{\mathcal{J}}(y(t), z(t), \tilde{u}(P_u, t), t) = \phi(y^N(t_f^N), z^N(t_f^N), t_f^N) + \sum_{i=1}^N \int_{t_0^i}^{t_f^i} \psi(y^i(t^i), z^i(t^i), \tilde{u}^i(P_{u^i}, t^i), t^i) \quad (51)$$

where $t^i \in [t_0^i, t_f^i]$ and $i = 1, \dots, N$.

9. The nonlinear optimization task: The objective function (50) is optimized subject to the modified DAE constraints (48) and (49). Then, the vector of the consisted initial conditions is needed to perform the numerical computations correctly. Additionally, the continuity and similarity constraints are considered.

The new nonlinear optimization task is formulated by the indicated components. The numerical continuous optimization procedure, which co-operates with the outer DAE solver, can be used to design an efficient sequential optimization algorithm.

10. The trajectories $y(t)$, $z(t)$, and $u(t)$, where:

$$y(t) = \begin{bmatrix} y_1^1(t^1) & \cdots & y_1^N(t^N) \\ \vdots & \ddots & \vdots \\ y_{n_y}^1(t^1) & \cdots & y_{n_y}^N(t^N) \end{bmatrix} \quad (52)$$

$$z(t) = \begin{bmatrix} z_1^1(t^1) & \cdots & z_1^N(t^N) \\ \vdots & \ddots & \vdots \\ z_{n_z}^1(t^1) & \cdots & z_{n_z}^N(t^N) \end{bmatrix} \quad (53)$$

$$\tilde{u}(P_u; t) = \begin{bmatrix} \tilde{u}_1^1(P_{u^1}; t^1) & \cdots & \tilde{u}_1^N(P_{u^N}; t^N) \\ \vdots & \ddots & \vdots \\ \tilde{u}_{n_y}^1(P_{u^1}; t^1) & \cdots & \tilde{u}_{n_y}^N(P_{u^N}; t^N) \end{bmatrix} \quad (54)$$

as well as the appropriate matrices P_{fi} , P_{gi} , P_{ui} and the vector of the initial conditions x_{y^i} for $i = 1, \dots, N$ are the solutions obtained by the new optimization algorithm. Moreover, it is suggested to save the final value of the similarity constraint vector. This information can be used to improve the similarity between the original and modified DAE constraints by a modification of the subinterval number N .

6. An Example of the Numerical Calculations

The designed algorithm is applied to optimize the heat and mass transfer phenomena described by a system of nonlinear differential-algebraic equations. In general, the technological effectiveness of the presented system in comparison with other available configurations was indicated in [25]. The modified counter-flow exchanger design task [26] with the DAE constraints in a semi-explicit form is constructed as follows:

$$\min_{y_1(0)} (y_1(l_f) - 30)^2 \quad (55)$$

subject to:

$$\begin{aligned} \dot{y}_1(l) &= -B \cdot (z_1(l) - y_1(l)) / l_f \\ \dot{y}_2(l) &= C \cdot (z_2(l) - y_2(l)) / l_f \\ \dot{y}_3(l) &= C \cdot (z_3(l) - y_3(l)) / l_f \\ 0 &= E_1 \cdot \dot{y}_1(l) - \dot{y}_2(l) - E_2 \dot{y}_3(l) \\ 0 &= z_2(l) - (z_1(l) - D \cdot (y_1(l) - z_1(l))) \\ 0 &= z_3(l) - 0.622 \cdot \frac{z_4(l)}{p_b - z_4(l)} \\ 0 &= z_4(l) - 6.107 \cdot e^{0.0726 \cdot z_2(l) - 2.912 \cdot 10^{-4} \cdot (z_2(l))^2 + 8.33 \cdot 10^{-7} \cdot (z_2(l))^3} \end{aligned} \quad (56)$$

where $y_1(l)$ —temperature in the first channel ($^{\circ}\text{C}$), $y_2(l)$ —temperature in the second channel ($^{\circ}\text{C}$), $y_3(l)$ —humidity ratio in the second channel (kg/kg), $z_1(l)$ —temperature of the plate surface in the first channel ($^{\circ}\text{C}$) $z_2(l)$ —temperature of the plate surface in the second channel, $z_3(l)$ —humidity ratio on the plate surface in the second channel (kg/kg), and $z_4(l)$ —the static pressure (h Pa). Moreover, there is a vector of global parameters:

$$p = \begin{bmatrix} B \\ C \\ D \\ E_1 \\ E_2 \\ P_b \end{bmatrix} = \begin{bmatrix} 30.0 \\ 30.0 \\ 0.058 \\ 1.0 \\ 2.5 \times 10^3 \\ 1000.0 \end{bmatrix} \quad (57)$$

as well as a vector of initial conditions:

$$y(0) = \begin{bmatrix} x_{y10} \\ 24.0 \\ 10.4 \cdot 10^{-3} \end{bmatrix} \quad (58)$$

and the independent variable domain:

$$l \in [0.0 \quad l_f] = [0.0 \quad 1.0]. \quad (59)$$

This model consisting of the differential-algebraic constraints has nonlinear algebraic relations. Therefore, analytical approaches are difficult to apply directly. Moreover, the numerical difficulty depends on the assumed parameters in the vector p . In particular, the values of B and C show how dynamic the mass and heat transfer is in the system. Then, according to the vector p , the solution procedure can result in an accurate solution for a slow dynamics or it can result in a computing error, if a considered transfer is rapid or sharp. The performed calculations indicate that the single shooting approach can be used if B and C are less than 10. The direct shooting strategies can be useful for B and C less than 20. For more dynamic processes, the co-operation of the multiple shooting with model parametrization seems to be appropriate.

To solve the presented optimization task, the multiple shooting approach with DAE constraint parametrization is implemented. The applied parametrization fulfills the conditions:

$$m_f = n_y \quad (60)$$

and:

$$m_g = n_z. \quad (61)$$

The algorithm is implemented with two numerical procedures from the MATLAB environment [27]:

- **fmincon**: The numerical optimization procedure is used to solve the parametrized continuous optimization task subject to the equality constraints. The numerical values of the objective and constraint functions are obtained with the outer DAE solver.
- **ode15s**: The outer solver is used to supply the state trajectories of the modified DAE relations for the given consistent initial conditions and values of parameters. This solver gives a support to finding consistent initial conditions.

The new algorithm with 10 equidistant subintervals results in a stable solution. The added constraints force the continuity of the state trajectories, which are presented in Figures 1–3. The solution obtained for each subinterval is marked with a different color. The initial value of $y_1(0) = 17.7556^{\circ}\text{C}$ is obtained. The value of the objective function is sequentially minimized, and finally, $(y_1(l_f) - 30)^2 \leq 10^{-6}$.

As was indicated previously, for the considered task with the parameter vector p (57), the classical single shooting-based approach cannot obtain any useful solution. Therefore, the efficient modifications of the multiple shooting method enable us to have a detailed insight into the course of the phenomena in a wide range of model parameters. The simulations can be performed even for such conditions, which seems to be difficult from the numerical calculations' perspective.

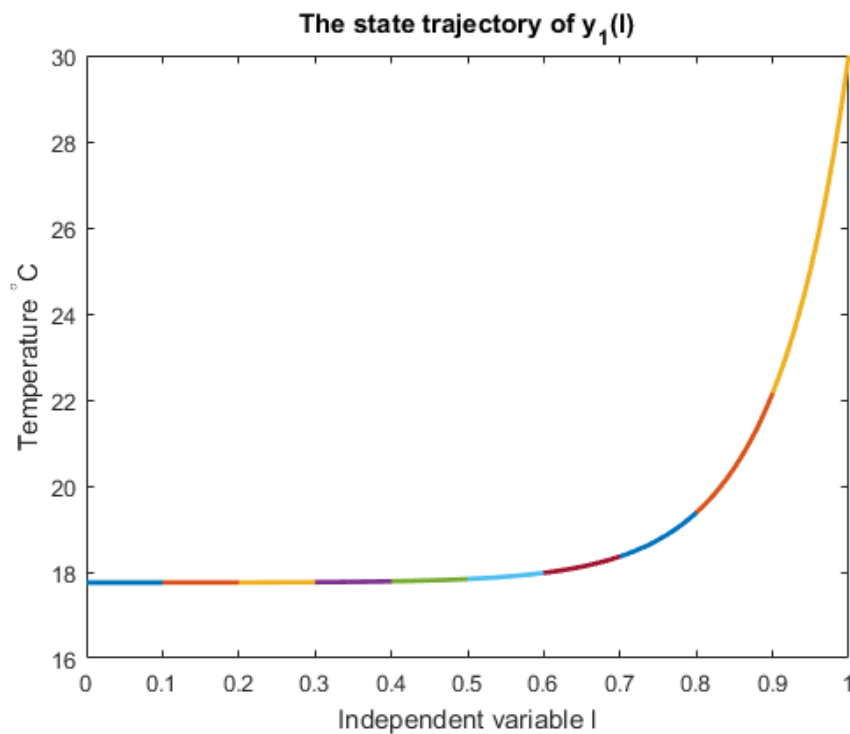


Figure 1. The trajectory of the state variable $y_1(t)$.

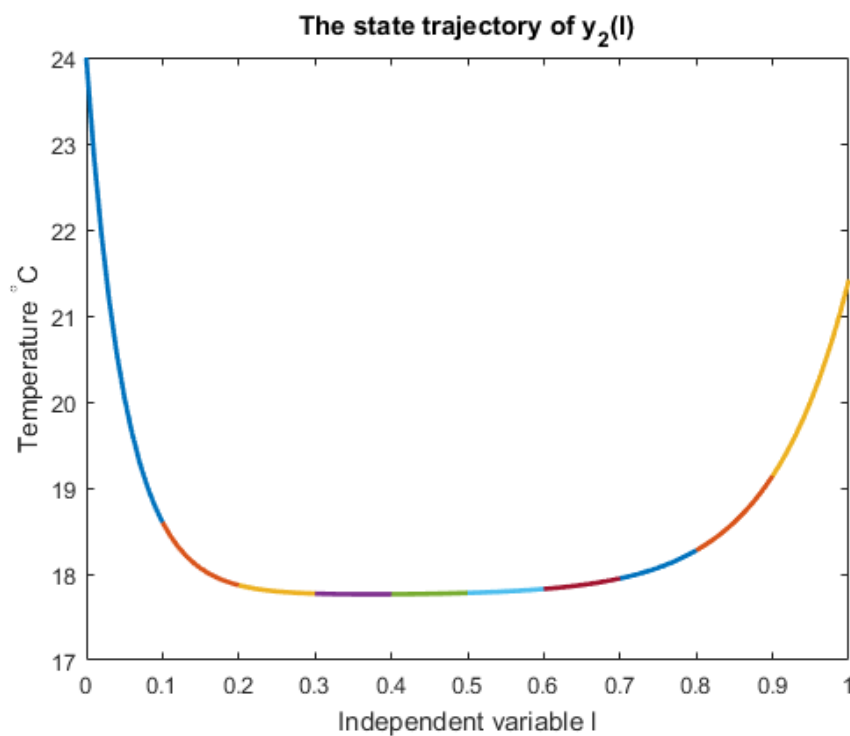


Figure 2. The trajectory of the state variable $y_2(t)$.

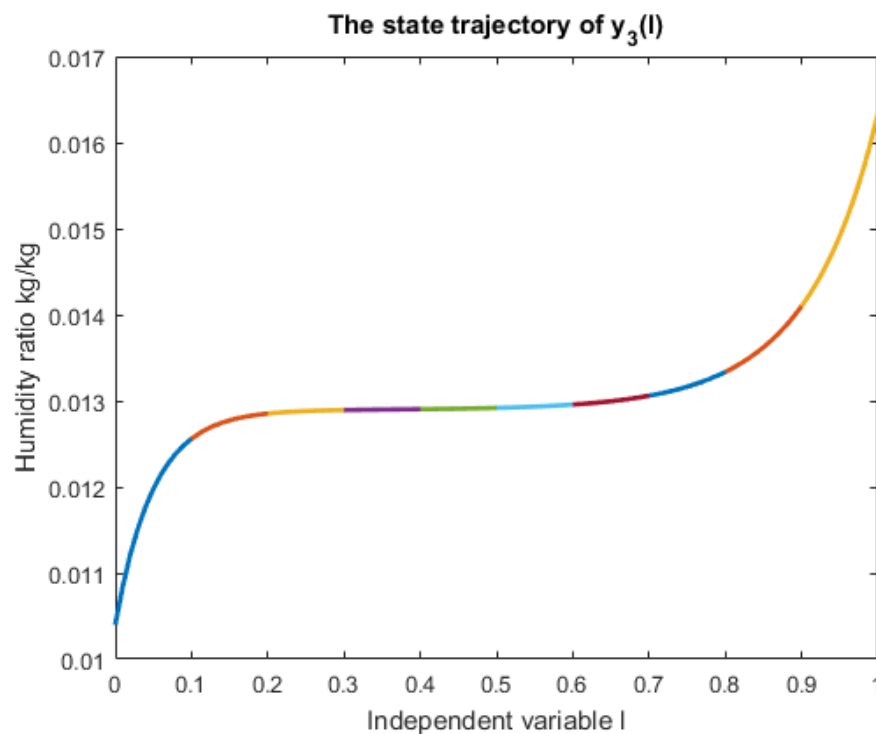


Figure 3. The trajectory of the state variable $y_3(t)$.

7. Conclusions

In the presented research, a new algorithm for the optimization task subject to the nonlinear differential-algebraic constraints is designed. The introduced methodology is an extension of the sequential approach for optimization, where the outer DAE solver can be used and the obtained state and control trajectories can be modified by the numerical optimization procedure step-by-step.

As a main result of this work, the new nonlinear optimization task formulation is derived. This formulation can be used and further analyzed in the context of the modern numerical optimization procedures with equality constraints. In other words, all parts of the nonlinear optimization task are clearly indicated and can be further modified and developed.

Much attention has been paid to the method of limiting the system dynamics in order to enable its numerical simulation. Therefore, the proposed method is based on a parametrization of the considered DAE constraints. The generalized parametrization approach is discussed in detail. Because of the system parametrization, during the optimization procedure, the similarity between the original and modified (parametrized) DAE constraints is measured. Moreover, the similarity constraints are considered as a part of the nonlinear optimization task.

The physical model parameters of heat and mass transfer systems have clear and known interpretations. In this work, a solution is sought for a such problem, which so far could not be solved for certain parameter values. Now, the values of the vector p can be modified according to a wide range of industrial applications. The simulation studies will enable the design of new mass and heat exchangers. In the presented case study, the solution is obtained with the co-operation of the numerical procedures `fmincon` and `ode15s` in the MATLAB environment. However, in fact, the choice of numerical methods is not imposed, but may be based on available procedures.

Some of the highlighted issues will be detailed in the future. Further work will be concentrated on issues related to special forms of the DAE model parametrization, as well as their properties and relations with the original dynamics. The extended similarity testing procedures will be considered also. The presented algorithm can find successful applications in other fields of engineering, like the design of rapid thermal processes in aviation, space, and nuclear energy.

Funding: This research was funded by the Department of Control Systems and Mechatronics at Wrocław University of Science and Technology.

Acknowledgments: The author would like to thank the anonymous reviewers for thoroughly reading the manuscript and providing valuable comments.

Conflicts of Interest: The author declares no conflict of interest. The founders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Beykal, B.; Onel, M.; Onel, O.; Pistikopoulos, E.N. A data driven optimization algorithm for differential algebraic equations with numerical infeasibilities. *AIChE J.* **2020**, *66*, e16657. [[CrossRef](#)] [[PubMed](#)]
2. Caspari, A.; Lüken, L.; Schäfer, P.; Vaupel, Y.; Mhamdi, A.; Biegler, L.T.; Mitsos, A. Dynamic optimization with complementarity constraints: Smoothing for direct shooting. *Comput. Chem. Eng.* **2020**, *139*, 106891. [[CrossRef](#)]
3. Pandelidis, D.; Drăg, M.; Drăg, P.; Worek, W.; Cetin, S. Comparative analysis between traditional and M-Cycle based cooling tower. *Int. J. Heat Mass Transf.* **2020**, *159*, 120124. [[CrossRef](#)]
4. Yang, C.; Gao, T.; Li, A.; Gao, X. Buoyancy-driven ventilation of an enclosure containing a convective area heat source. *Int. J. Therm. Sci.* **2021**, *159*, 106551. [[CrossRef](#)]
5. Guan, B.; Liu, X.; Zhang, T. Energy performance analysis on segmented liquid desiccant air-conditioning system for bus spray-paint booths. *J. Clean. Prod.* **2021**, *278*, 123898. [[CrossRef](#)]
6. El Loubani, M.; Ghaddar, N.; Ghali, K.; Itani, M. Hybrid cooling system integrating PCM-desiccant dehumidification and personal evaporative cooling for hot and humid climates. *J. Build. Eng.* **2021**, *33*, 101580. [[CrossRef](#)]
7. Schreiber, T.; Eschweiler, S.; Baranski, M.; Müller, D. Application of two promising Reinforcement Learning algorithms for load shifting in a cooling supply system. *Energy Build.* **2020**, *229*, 110490. [[CrossRef](#)]
8. Ivanko, D.; Sørensen, Å.L.; Nord, N. Selecting the model and influencing variables for DHW heat use prediction in hotels in Norway. *Energy Build.* **2020**, *228*, 110441. [[CrossRef](#)]
9. Kelley, M.T.; Baldick, R.; Baldea, M. A direct transcription-based multiple shooting formulation for dynamic optimization. *Comput. Chem. Eng.* **2020**, *140*, 106846. [[CrossRef](#)]
10. Fritzson, P. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*; Wiley-IEEE Press: Piscataway, NJ, USA, 2004.
11. Ploch, T.; von Lieres, E.; Wiechert, W.; Mitsos, A.; Hannemann-Tamás, R. Simulation of differential-algebraic equation systems with optimization criteria embedded in Modelica. *Comput. Chem. Eng.* **2020**, *140*, 106920. [[CrossRef](#)]
12. Lamour, R. Index determination and calculation of consistent initial values for DAEs. *Comput. Math. Appl.* **2005**, *50*, 1125–1140. [[CrossRef](#)]
13. Resende, M.; Ratusznei, S.M.; Torres Suazo, C.A.; Giordano, R.C.; Simulating a ceramic membrane bioreactor for the production of penicillin: An example of the importance of consistent initialization for solving DAE systems. *Process. Biochem.* **2002**, *37*, 1297–1305. [[CrossRef](#)]
14. Vieira, R.C.; Biscaia, E.C., Jr. Direct methods for consistent initialization of DAE systems. *Comput. Chem. Eng.* **2001**, *25*, 1299–1311. [[CrossRef](#)]
15. Biscaia, E.C., Jr.; Vieira, R.C. Heuristic optimization for consistent initialization of DAEs. *Comput. Chem. Eng.* **2000**, *24*, 2183–2191. [[CrossRef](#)]
16. Brenan, K.E.; Campbell, S.L.; Petzold, L.R. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1996.
17. Betts, J.T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2010.
18. Caruso, A.; Quarta, A.A.; Mengali, G. Comparison between direct and indirect approach to solar sail circle-to-circle orbit raising optimization. *Astrodynamics* **2019**, *3*, 273–284. [[CrossRef](#)]
19. Drăg, P.; Styczeń, K. A chain smoothing Newton method for heat and mass transfer control with discrete variability DAE models. *Int. Commun. Heat Mass Transf.* **2021**, *120*, 105056. [[CrossRef](#)]

20. Banga, J.R.; Balsa-Canto, E.; Moles C.G.; Alonso A.A. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *J. Biotechnol.* **2005**, *117*, 407–419. [[CrossRef](#)] [[PubMed](#)]
21. Diedam, H.; Sager, S. Global optimal control with the direct multiple shooting method. *Optim. Control Appl. Methods* **2018**, *39*, 449–470. [[CrossRef](#)]
22. Drąg, P. *Algorytmy Sterowania Wielostadialnymi Procesami Deskryptorowymi (The Control Algorithms for Multi-Stage Descriptor Systems)*; Akademicka Oficyna Wydawnicza EXIT: Warszawa, Poland, 2016.
23. Biegler, L.T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*; SIAM: Philadelphia, PA, USA, 2010.
24. Kelley, C.T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1995.
25. Pandelidis, D.; Anisimov, S.; Worek, W.M. Comparison study of the counter-flow regenerative evaporative heat exchangers with numerical methods. *Appl. Therm. Eng.* **2015**, *84*, 211–224. [[CrossRef](#)]
26. Pandelidis, D.; Anisimov, S.; Worek, W.M. Performance study of counter-flow indirect evaporative air coolers. *Energy Build.* **2015**, *109*, 53–64. [[CrossRef](#)]
27. Higham, D.J.; Higham, N.J. *MATLAB Guide*; SIAM: Philadelphia, PA, USA, 2016.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).