Reachability Analysis of Nonlinear Differential-Algebraic Systems

Matthias Althoff and and Bruce Krogh, *Fellow, IEEE*

***Abstract*—This paper presents a numerical procedure for the reachability analysis of systems with nonlinear, semi-explicit, index-1 differential-algebraic equations. The procedure computes reachable sets for uncertain initial states and inputs in an overapproximative way, i.e. it is guaranteed that all possible trajectories of the system are enclosed. Thus, the result can be used for formal verification of system properties that can be specified in the state space as unsafe or goal regions. Due to the representation of reachable sets by zonotopes and the use of highly scalable operations on them, the presented approach scales favorably with the number of state variables. This makes it possible to solve problems of industry-relevant size, as demonstrated by a transient stability analysis of the IEEE 14-bus benchmark problem for power systems.**

***Index Terms*—Reachability analysis, formal safety verification, nonlinear differential-algebraic equations (DAEs), zonotopes, power systems.**

I. Introduction

For many model-based control problems, it is not sufficient to check properties of a dynamic system by simulations of single trajectories, e.g., when it is required to verify if specifications are not violated for all possible initial states, disturbances, and parameters. Computing the set of all solutions is often referred to as reachability analysis, which evolved from extensions of algorithms for the analysis of graphs [43] to discrete systems [36], timed automata [7], and eventually to systems with continuous and hybrid (mixed discrete-continuous) dynamics [8]. The paper presents a numerical procedure for the reachability analysis of systems with differential-algebraic equations (DAEs), a class of systems that has received only limited attention in the reachability analysis literature [18], [19], [38].

DAEs occur in many practical applications, typically when there are constraints on the state variables [10], [14]. State variable constraints occur, for example, in robotics, when a robot has to move its end-effector along a surface, or in electrical networks, when currents in a node are constraint by Kirchhoff's law. DAE systems also arise from the application of model-order reduction and singular perturbation techniques of ODEs [14, Chapter 1.3.3], and discretization of partial differential equations (PDEs) [14, Chapter 1.3.4].

Models for dynamic systems are typically derived in the implicit form $0 = F(\dot{\tilde{x}}, \tilde{x}, u, t)$, where $\tilde{x} \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, and t is the time, which is explicit when the system is time-varying. If $\partial f(\dot{\tilde{x}}, \tilde{x}, u, t)/\partial \dot{\tilde{x}}$ is non-singular, one can rewrite the implicit form to an explicit ordinary differential equation (ODE) of the form $\dot{\tilde{x}} = \tilde{f}(\tilde{x}, u, t)$ in most cases [10, Chapter 1.3]. Otherwise, the system model is a set of differential-algebraic equations of the form $0 = f(\dot{x}, x, y, u, t)$, where x is separated into a vector $x \in \mathbb{R}^{n_d}$ of so-called *differential variables* for which a derivative is present, and a vector $y \in \mathbb{R}^{n_a}$ of so-called *algebraic variables* for which no derivative is present. This formalism is also the basic representation for acausal modeling [21].

In this paper we consider time-invariant, semi-explicit, index-1 DAEs, which is the most common class of DAEs for practical problems. Additionally, in most cases, one can apply index reduction techniques in order to obtain index-1 DAEs [22], [24]. Some software packages for engineering problems are capable of solving only index-1 DAEs, such as the ode15s-solver in MATLAB [47] or DASSL in Dymola [41]. The approach presented in this paper can also be applied to nonlinear ordinary differential equations as a special case of DAEs.

As mentioned above, there is only a small amount of literature on reachability analysis for DAEs, especially on techniques that scale well with the number of differential and algebraic variables. All of the current literature on reachability analysis for DAEs focuses on index-1 systems. Most of the work on reachability analysis of DAEs has been done using level-set methods [18], [38]. These methods reformulate the reachability problem to solving Hamilton-Jacobi partial differential equations, which is done by discretizing the state space. As a consequence, the computational complexity is exponential in the number of state variables, which typically limits the application to systems with no more than four continuous variables. Besides level-set methods, Dang et al. investigated DAEs for electrical circuits using polyhedral set representations [19]. This method scales more favorably with the number of state variables compared to level-set methods, but requires projections of the reachable set onto the constraint manifold determined by the algebraic equations. This projection is computationally expensive and it is not guaranteed that the computed approximation of the reachable set projection onto the manifold is an overapproximation.

A problem similar to reachability analysis is addressed in guaranteed integration, where one guarantees the enclosure of a solution despite rounding errors, typically computed for a small region around a single initial state without considering uncertain time-varying inputs. There are many approaches

Matthias Althoff is with the Department of Computer Science, Technische Universität München, 85748 Garching, Germany, email: althoff@tum.de

Bruce H. Krogh is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, email: krogh@ece.cmu.edu

Manuscript received month day, year; revised month day, year.

for guaranteed integration of ODEs and hybrid systems with ODEs as continuous dynamics (see e.g. [40], [44]), while the literature for DAEs is dominated by a rigorous Taylor series approach [29], [30].

The reachability computation proposed in this work does not require a projection operation onto the constraint manifold as often performed in numerical solvers of DAEs. This is advantageous since projection of a partial solution for a small time increment onto the manifold (algebraic variables are assumed to be constant for the time increment), results in approximate solutions whose distance to the exact solution is hard to rigorously quantify. Additionally, projections of sets are typically only feasible when the manifold is a hyperplane. We compute the reachable set of the differential variables first, but based on this result, obtain the reachable set of the algebraic variables without using projection. The reachable set of the differential variables is computed for short consecutive time intervals by abstracting the original nonlinear dynamics to a linear system with set-valued right-hand side (a *differential inclusion* [11], [48]). For reachability analysis it is equivalent to consider a system of differential equations for which inputs and/or parameters are uncertain within sets or the corresponding system of differential inclusions, where most publications do not include the term *differential inclusion* in their title (except e.g. [13]). Our method scales with $\mathcal{O}(n^5)$, where n is number of differential and algebraic variables, which can be reduced to $\mathcal{O}(n^3)$ for mild nonlinearities. The worst-case complexity holds under the assumption that the reachable set does not have to be split, which might be required when the set of initial conditions, or the nonlinearity measure is large. In that case, the complexity is $\mathcal{O}(2^{\tilde{n}}n^5)$, where $\tilde{n}$ is the number of variables occurring in nonlinear terms. The low complexity makes it possible to verify properties of DAE systems with sizes relevant in practice. As an example, we consider the problem of showing that after an intermittent power drop-out of a power plant, the initial operating condition of the power grid is restored for all possible initial states, which is called *transient stability analysis* in the power system literature [34]. The considered problem is rather large with 14 differential and 28 algebraic variables, summing up to a total of 42 continuous state variables.

Obviously, the proposed approach can also be applied to nonlinear ODEs. Most other approaches for nonlinear system reachability also simplify the dynamics, either within regions of a fixed state space partition [9], [42], or by simplification in the vicinity of the reachable set [20], [28], which is the approach used in the previous work [6]. The latter approach generally outperforms fixed partitions, which suffer from (1) the exponential growth of regions with respect to the number of state variables, and (2) the required intersection operations of hybrid system reachability analysis. Approaches which do not use abstraction are mostly based on optimization techniques, which are computationally more expensive [17], [37], [50].

The paper is organized as follows. In Sec. II, we formalize the reachability problem for systems with DAEs. In Sec. III, we recapitulate the computation of reachable sets for linear differential inclusions and operations on zonotopes, which are used for representing the reachable sets. For computing reachable sets of DAEs in an overapproximative way, the original dynamics are abstracted to linear differential inclusions using the conservative linearization approach in Sec. IV so that well-known techniques from Sec. III can be applied. The abstraction requires the computation of the linearization error, which is addressed in Sec. V. A summary of the newly developed algorithm is presented in Sec. VI, which is applied to the transient stability analysis of the IEEE 14-bus system in Sec. VII, followed by the conclusion in Sec. VIII.

II. Problem Formulation

We consider time-invariant, semi-explicit, index-1 DAEs without parametric uncertainties. We do not consider parametric uncertainties in order to focus on the novelties of the paper; the extension to parametric uncertainties can be done using the methods presented in [6]. Using the previously introduced vectors of differential variables x, algebraic variables y, and inputs u, the semi-explicit DAE can generally be written as

$$\begin{aligned} \dot{x} &= f(x(t), y(t), u(t)) \\ 0 &= g(x(t), y(t), u(t)), \\ [x^T(0), y^T(0)]^T &\in \mathcal{R}(0), \quad u(t) \in \mathcal{U}, \end{aligned} \tag{1}$$

where $\mathcal{R}(0)$ overapproximates the set of consistent initial states and $\mathcal{U}$ is the set of possible inputs.

The initial state is consistent when $g(x(0), y(0), u(0)) = 0$, while for DAEs with an index greater than 1, further hidden algebraic constraints have to be considered [10, Chapter 9.1]. For an implicit DAE, the index-1 property holds if and only if $\forall t : \det(\frac{\partial g(x(t),y(t),u(t))}{\partial y}) \neq 0$, i.e. the Jacobian of the algebraic equations is non-singular [14, p. 34]. Loosely speaking, the index specifies the distance to an ODE (which has index 0) by the number of required time differentiations of the general form $0 = F(\dot{\tilde{x}}, \tilde{x}, u, t)$ along a solution $\tilde{x}(t)$, in order to determine $\dot{\tilde{x}}$ as a continuous function of $\tilde{x}, t$ [10, Chapter 9.1].

We assume that (1) has a unique solution (see [14, Def. 2.2.1]) denoted by $\gamma(t, x(0), y(0), u(\cdot))$ for all consistent initial states $x(0) \in \mathbb{R}^{n_d}$, $y(0) \in \mathbb{R}^{n_a}$, where $u(\cdot)$ refers to a piecewise continuous input trajectory, rather than an input at a specific point in time. The objective is to find the set of reachable states of (1) over some time horizon $t \in [0, t_f]$, which is defined as

$$\mathcal{R}^e([0,t_f]) := \Big\{ \gamma(t, x(0), y(0), u(\cdot)) \Big| [x^T(0), y^T(0)]^T \in \mathcal{R}(0), \\ u(t) \in \mathcal{U}, \, t \in [0, t_f] \Big\}.$$

The superscript e on $\mathcal{R}^e([0,t_f])$ denotes the exact reachable set, which cannot be computed for nonlinear DAE systems [35]. For this reason, we aim to compute overapproximations $\mathcal{R}([0,t_f]) \supseteq \mathcal{R}^e([0,t_f])$ which are as accurate as possible, while at the same time ensuring that the computations are efficient and scale well with the system dimension $n = n_d + n_a$ (n_d: number of differential variables, n_a: number of algebraic variables.). From now on, we often only say *reachable set* when referring to an *overapproximative reachable set* to simplify the wording. The projection of the reachable set onto the differential variables is denoted by $\mathcal{R}^d([0,t_f])$ and by $\mathcal{R}^a([0,t_f])$ when projected onto the algebraic variables.

III. Preliminaries

The approach presented in this work is based on known techniques for computing reachable sets of linear differential inclusions, which are recapitulated in this section. We also recapitulate well-known operations on zonotopes, which are chosen as the set representation due to their good performance for required operations in reachability analysis of linear differential inclusions. As presented later, zonotopes are also a good choice for newly proposed overapproximations of nonlinear operations.

A. Reachable Set Computation of Linear Systems

Reachable set computations are typically performed iteratively by computing the reachable set of short time intervals $t \in \tau_k := [t_k, t_{k+1}]$. In this work, we restrict ourselves to constant-size time intervals with $t_k := k\,r$ to focus on the main innovations, where $k \in \mathbb{N}$ is the time step and $r \in \mathbb{R}^+$ is referred to as the time increment or step size. An extension to variable step sizes is described in [23]. The reachable set for a specified time horizon $t_f \in \mathbb{R}^+$ is stored as a list of reachable sets $\mathcal{R}(\tau_k)$ until $t_{k+1} \geq t_f$. In order to compute reachable sets of time intervals, the reachable sets of points in time $\mathcal{R}(t_k)$ are computed as well in this work. From now on, we focus on computing the iterative solution for the next point in time and the next time interval.

The iterative computation of reachable sets for linear systems requires set-based addition (*Minkowski addition*) and set-based multiplication:

$$\begin{aligned}\mathcal{X} \oplus \mathcal{Y} &:= \{x + y | x \in \mathcal{X}, y \in \mathcal{Y}\},\\ \mathcal{X} \otimes \mathcal{Y} &:= \{x\,y | x \in \mathcal{X}, y \in \mathcal{Y}\}.\end{aligned}$$

Note that the symbol for set-based multiplication is often omitted for simplicity of notation, and that one or both operands can be singletons. The following presents a brief description of the main steps for obtaining reachable sets for a single time interval.

Given is the linear differential inclusion $\dot{\tilde{x}} \in \tilde{A}\tilde{x}(t) \oplus \tilde{\mathcal{U}}$, where $\tilde{x} \in \mathbb{R}^{n_d}$, $\tilde{A} \in \mathbb{R}^{n_d \times n_d}$, $\tilde{\mathcal{U}} \subset \mathbb{R}^{n_d}$ is a set of uncertain inputs. We use a tilde for the variables of the linear differential inclusion to distinguish the variables from the ones of the original nonlinear DAEs. For further computations, we introduce the center u_c and the deviation from the center $\tilde{\mathcal{U}}_\Delta := \tilde{\mathcal{U}} \oplus (-u_c)$ of $\tilde{\mathcal{U}}$. According to [2], the reachable set for a time interval τ_k is computed as shown in Fig. 1:

1) Starting from $\mathcal{R}^d(t_k)$, compute the set of all solutions $\mathcal{R}^d_h(t_{k+1})$ for the affine dynamics $\dot{\tilde{x}} = \tilde{A}\tilde{x}(t) + u_c$ at time t_{k+1}.
2) Obtain the convex hull of $\mathcal{R}^d(t_k)$ and $\mathcal{R}^d_h(t_{k+1})$ to approximate the reachable set for the time interval τ_k.
3) Compute $\mathcal{R}^d(\tau_k)$ by enlarging the convex hull to first bound all affine solutions within τ_k and secondly account for the set of uncertain inputs $\tilde{\mathcal{U}}_\Delta$.

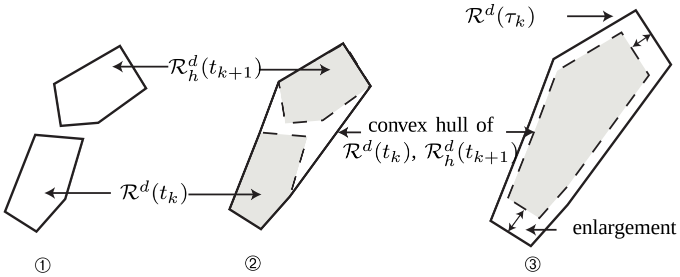

$\mathcal{R}^d(\tau_k)$
$\mathcal{R}^d_h(t_{k+1})$
convex hull of
$\mathcal{R}^d(t_k)$, $\mathcal{R}^d_h(t_{k+1})$
$\mathcal{R}^d(t_k)$
enlargement
①
②
③

Fig. 1. Steps for the computation of an overapproximation of the reachable set for a linear system.

Using $r = t_{k+1} - t_k$, the well-known solution of $\mathcal{R}^d_h(t_{k+1})$ is

$$\mathcal{R}^d_h(t_{k+1}) = e^{\tilde{A}r}\mathcal{R}^d(t_k) + \underbrace{\int_0^r e^{\tilde{A}(r-t)}\,dt\,u_c}_{=:x_p(r)}.$$

If $\tilde{A}$ is invertible, $x_p(r)$ can be computed as $\tilde{A}^{-1}(e^{\tilde{A}r} - I)u_c$, where I is the identity matrix. However, since $\tilde{A}$ is not always invertible, we compute $x_p(r)$ by integrating the Taylor series of $e^{\tilde{A}r} = \sum_{i=0}^{\infty}(\tilde{A}r)^i/(i!)$:

$$\begin{aligned}x_p(r) =& \Big(\sum_{i=0}^{\eta} \frac{\tilde{A}^i r^{i+1}}{(i+1)!} + \underbrace{\sum_{i=\eta+1}^{\infty} \frac{\tilde{A}^i r^{i+1}}{(i+1)!}}_{=:E_p(r)}\Big)u_c\\ &\in \underbrace{\Big(\sum_{i=0}^{\eta} \frac{\tilde{A}^i r^{i+1}}{(i+1)!} \oplus \mathcal{E}_p(r)\Big)}_{=:\Gamma(r)}\,u_c,\end{aligned}$$

The remainder $E_p(r)$ can be overapproximated by an interval matrix $E_p(r) \in \mathcal{E}_p(r) := [-W(r)\,r, W(r)\,r]$, i.e., by a matrix with lower and upper bounds on each element. Using symmetric bounds on $E_p(r)$, these bounds can be obtained from

$$\begin{aligned}|E_p(r)| =& \Big|\sum_{i=\eta+1}^{\infty} \frac{\tilde{A}^i}{(i+1)!} r^{i+1}\Big| \leq \sum_{i=\eta+1}^{\infty} \frac{|\tilde{A}|^i r^{i+1}}{(i+1)!}\\ \leq& \Big(\sum_{i=\eta+1}^{\infty} \frac{|\tilde{A}|^i r^i}{i!}\Big) r = \underbrace{\Big(e^{|\tilde{A}|r} - \sum_{i=0}^{\eta} \frac{|\tilde{A}|^i r^i}{i!}\Big)}_{=:W(r)} r. \qquad (2)\end{aligned}$$

Next, we discuss the enlargement of the convex hull denoted by $\mathcal{R}^d_\epsilon$ to contain all affine solutions for τ_k (the construction of the convex hull is presented below in Sec. III-B). According to [2, Chap. 3.2], the solution is obtained using $\tilde{W}(r) := W(r)r$ and $W(r)$ from (2):

$$\begin{aligned}\mathcal{R}^d_\epsilon &:= \big(\mathcal{F} \otimes \mathcal{R}^d(t_k)\big) \oplus \big(\tilde{\mathcal{F}} \otimes u_c\big)\\ \mathcal{F} &:= \left(\bigoplus_{i=2}^{\eta} \left[\left(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\right) r^i, 0\right] \frac{\tilde{A}^i}{i!}\right) \oplus [-W(r), W(r)]\\ \tilde{\mathcal{F}} &:= \left(\bigoplus_{i=2}^{\eta+1} \left[\left(i^{\frac{-i}{i-1}} - i^{\frac{-1}{i-1}}\right) r^i, 0\right] \frac{\tilde{A}^{i-1}}{i!}\right) \oplus [-\tilde{W}(r), \tilde{W}(r)]\end{aligned}$$

The reachable set due to the uncertain input $\tilde{\mathcal{U}}_\Delta$ is obtained as derived in [5]:

$$\mathcal{R}^d_p(\tilde{\mathcal{U}}_\Delta, r) := \bigoplus_{i=0}^{\eta} \left(\frac{\tilde{A}^i\, r^{i+1}}{(i+1)!} \tilde{\mathcal{U}}_\Delta \right) \oplus \big([-\tilde{W}(r), \tilde{W}(r)] \otimes |\tilde{\mathcal{U}}_\Delta|\big), \tag{3}$$

where the absolute value of a set of matrices $\mathcal{M}$ is defined elementwise as $|\mathcal{M}|_{ij} := \sup\big\{|m_{ij}| \big| m \in \mathcal{M}\big\}$, which equivalently applies to the vector set $\tilde{\mathcal{U}}_\Delta$. Note that when $\tilde{\mathcal{U}}_\Delta$ is not convex, one has to compute with the convex hull of the input set in order to apply the above formula, see [5].

The reachable sets for the next point in time and time interval are obtained by combining all previous results (see [5]):

$$\begin{aligned} \mathcal{R}^d(t_{k+1}) :=& e^{Ar}\mathcal{R}^d(t_k) \oplus \Gamma(r) u_c \oplus \mathcal{R}^d_p(\tilde{\mathcal{U}}_\Delta, r), \\ \mathcal{R}^d(\tau_k) :=& \mathrm{CH}\big(\mathcal{R}^d(t_k), e^{Ar}\mathcal{R}^d(t_k) \oplus \Gamma(r) u_c\big) \\ & \oplus \mathcal{R}^d_\epsilon \oplus \mathcal{R}^d_p(\tilde{\mathcal{U}}_\Delta, r), \end{aligned} \tag{4}$$

where $\mathrm{CH}()$ returns the convex hull. The representation of the reachable set by zonotopes is addressed in the next section.

B. Representation of Reachable Sets by Zonotopes

As shown above, the set operations required for reachability analysis of linear systems are matrix and interval matrix multiplication, Minkowski addition, absolute value computation, and convex hull. All of these can be efficiently computed using zonotopes, which makes zonotopes very attractive for reachability computations of linear systems [25], [27]. Besides zonotopes, support functions have been shown to be useful for computing reachable sets for linear systems when using a wrapping-free computation scheme [26]. However, there exists no wrapping-free algorithm for nonlinear systems, so we use zonotopes since they can be used for efficient nonlinear reachability analysis, too.

Definition 1 (Zonotope) *Given a center $c \in \mathbb{R}^n$ and so-called generators $g^{(i)} \in \mathbb{R}^n$, a zonotope is defined as*

$$\mathcal{Z} := \Big\{ x \in \mathbb{R}^n \Big| x = c + \sum_{i=1}^{p} \beta_i g^{(i)}, \beta_i \in [-1, 1] \Big\}$$

We write in short $\mathcal{Z} = (c, g^{(1)}, \dots, g^{(p)})$ and define the order of a zonotope as $\rho := \frac{p}{n}$, where p is the number of generators.

The multiplication with a matrix $M \in \mathbb{R}^{o \times n}$ and the Minkowski addition of two zonotopes $\mathcal{Z}_1 = (c,\ g^{(1)},\ \dots,\ g^{(p_1)})$ and $\mathcal{Z}_2 = (d,\ h^{(1)},\ \dots,\ h^{(p_2)})$, are a direct consequence of the zonotope definition (see [33]):

$$\begin{aligned} &\mathcal{Z}_1 \oplus \mathcal{Z}_2 = (c + d, g^{(1)}, \dots, g^{(p_1)}, h^{(1)}, \dots, h^{(p_2)}) \\ &M \otimes \mathcal{Z}_1 = (M\, c, M\, g^{(1)}, \dots, M\, g^{(p_1)}) \end{aligned} \tag{5}$$

We additionally require the convex hull of $\mathcal{Z}_1$ and $e^{Ar}\mathcal{Z}_1$ (see [25]):

$$\begin{aligned} &\mathrm{CH}(\mathcal{Z}_1, e^{Ar}\mathcal{Z}_1) \subseteq \\ &\frac{1}{2}(c_1 + e^{Ar}c_1, g^{(1)} + e^{Ar}g^{(1)}, \dots, g^{(p_1)} + e^{Ar}g^{(p_1)}, \\ &\quad c_1 - e^{Ar}c_1, g^{(1)} - e^{Ar}g^{(1)}, \dots, g^{(p_1)} - e^{Ar}g^{(p_1)}). \end{aligned} \tag{6}$$

For the multiplication with an interval matrix $\mathcal{M}$, we split $\mathcal{M}$ into a real-valued matrix $M \in \mathcal{R}^{n \times n}$ and an interval matrix with radius $S \in \mathcal{R}^{n \times n}$, such that $\mathcal{M} = M \oplus [-S, S]$. After introducing S_j as the j^{th} row of S, the result is overapproximated as shown in [2, Theorem 3.3] by

$$\begin{aligned} \mathcal{M}\mathcal{Z}_1 \subseteq& (M\mathcal{Z}_1 \oplus [-S, S]\mathcal{Z}_1) \\ \subseteq& (Mc_1, Mg^{(1)}, \dots, Mg^{(p_1)}, h^{(1)}, \dots, h^{(n)}) \\ h^{(i)}_j =& \begin{cases} S_j(|c| + \sum_{k=1}^{p_1} |g|^{(k)}), \text{ for } i = j \\ 0, \text{ for } i \neq j \end{cases} . \end{aligned}$$

We will also need the enclosure of a zonotope by a multidimensional box [2, Prop. 2.2] and its absolute value:

$$\begin{aligned} \mathrm{box}(\mathcal{Z}_1) :=& [c_1 - \Delta g, c_1 + \Delta g], \qquad \Delta g := \sum_{i=1}^{p_1} |g^{(i)}|, \\ |\mathcal{Z}_1| :=& |c_1| + \Delta g. \end{aligned} \tag{7}$$

The representation of reachable sets with zonotopes allows an efficient computation as presented later.

IV. Conservative Linearization

To apply the methods presented in the previous section to compute reachable sets for DAEs, an abstraction of the original nonlinear DAEs to linear differential inclusions is performed for each consecutive time interval τ_k of the reachable set computation (see Sec. III-A). A different abstraction is used for each time interval to minimize the overapproximation error. We first discuss the conservative linearization procedure, followed by the linearization error handling.

A. Linearization Procedure

For a concise notation of the conservative linearization, we introduce $z := [x^T, y^T, u^T]^T$, the linearization point $z^* := [x^{*T}, y^{*T}, u^{*T}]^T$, and $\mathcal{R}^z := \mathcal{R}(\tau_k) \times \mathcal{U}$. The linearization point for the differential variables is chosen for each iteration close to the center of the next reachable set $\mathcal{R}(\tau_k)$, which is a good heuristic for minimizing the linearization error. The Euler integration method is used for the time increment $0.5r$ to approximate this point by $x^* = c^d + 0.5r \cdot f(c^d, c^a, c^u)$, where c^d, c^a, c^u are the volumetric centers of the sets $\mathcal{R}^d(t_k)$, $\mathcal{R}^a(t_k)$, and $\mathcal{U}$. We choose $u^* = c^u$ and the linearization point of the algebraic part is obtained by solving $0 = g(x^*, y^*, u^*)$ using a Newton-Raphson algorithm.

The linearization of (1) is performed using a first-order Taylor expansion with Lagrangian remainder:

$$\begin{aligned} &\dot{x}_i = f_i(z(t)) \in f_i(z^*) + \frac{\partial f_i(z)}{\partial z}\Big|_{z=z^*}(z(t) - z^*) \\ &\oplus \underbrace{\left\{ \frac{1}{2}(z(t) - z^*)^T \frac{\partial^2 f_i(z)}{\partial z^2}\Big|_{z=\xi}(z(t) - z^*) \Big| \xi,\ z(t) \in \mathcal{R}^z \right\}}_{=:\mathcal{L}^d_i}, \\ &0 = g_j(z(t)) \in g_j(z^*) + \frac{\partial g_j(z)}{\partial z}\Big|_{z=z^*}(z(t) - z^*) \\ &\oplus \underbrace{\left\{ \frac{1}{2}(z(t) - z^*)^T \frac{\partial^2 g_j(z)}{\partial z^2}\Big|_{z=\xi}(z(t) - z^*) \Big| \xi,\ z(t) \in \mathcal{R}^z \right\}}_{=:\mathcal{L}^a_j}, \end{aligned} \tag{8}$$

where $\mathcal{L}^d_i$ denotes the projection of $\mathcal{L}^d$ onto the i^{th} coordinate. The Lagrangian remainders $\mathcal{L}^d, \mathcal{L}^a$ enclose all higher-order terms if ξ can take any value of the linear combination of z and z^*, i.e. $\xi \in \{\alpha z + (1-\alpha) z^* | \alpha \in [0,1]\}$, which follows from the mean value theorem [12, p. 87]. Since for the time interval τ_k, (1) $z(t)$ can take any values from $\mathcal{R}^z$, (2) $\mathcal{R}^z$ is represented by a convex zonotope, and (3) z^* is chosen as an interior point of this set, it follows that for $\xi \in \mathcal{R}^z$ the set of Lagrangian remainders is captured.

For subsequent derivations, it is required to separate the effects from differential variables, algebraic variables, and inputs. Thereto, we define the following submatrices of the Jacobians:

$$\frac{\partial f(z)}{\partial z}\Big|_{z=z^*} = [A,\, C,\, B], \quad \frac{\partial g(z)}{\partial z}\Big|_{z=z^*} = [D,\, F,\, E], \quad (9)$$

where $A \in \mathbb{R}^{n_d \times n_d}$, $B \in \mathbb{R}^{n_d \times m}$, $C \in \mathbb{R}^{n_d \times n_a}$, $D \in \mathbb{R}^{n_a \times n_d}$, $E \in \mathbb{R}^{n_a \times m}$, $F \in \mathbb{R}^{n_a \times n_a}$, and n_d, n_a, m are the number of differential, algebraic, and input variables, respectively. Inserting the abbreviation $z = [x^T, y^T, u^T]^T$ and the matrices A-F into the Taylor expansion (8), and introducing $H^{d,(i)}(\xi) := \frac{\partial^2 f_i(z)}{\partial z^2})\big|_{z=\xi}$, $H^{a,(j)}(\xi) := \frac{\partial^2 g_j(z)}{\partial z^2})\big|_{z=\xi}$, $\mathcal{R}^z_\Delta := \mathcal{R}^z \oplus (-z^*)$, $\nu(t) := z(t) - z^*$, yields

$$\begin{aligned}\dot{x} \in& f(z^*) + A(\underbrace{x(t)-x^*}_{=:\Delta x(t)}) + B(\underbrace{u(t)-u^*}_{=:\Delta u(t)}) + C(\underbrace{y(t)-y^*}_{=:\Delta y(t)}) \\ &\oplus \Big\{\frac{1}{2}\sigma \Big| \sigma_i = \nu^T H^{d,(i)}(\xi)\nu,\, \xi \in \mathcal{R}^z,\, \nu \in \mathcal{R}^z_\Delta\Big\}, \quad (10)\\ 0 \in& g(z^*) + D(\underbrace{x(t)-x^*}_{=:\Delta x(t)}) + E(\underbrace{u(t)-u^*}_{=:\Delta u(t)}) + F(\underbrace{y(t)-y^*}_{=:\Delta y(t)}) \\ &\oplus \Big\{\frac{1}{2}\phi \Big| \phi_j = \nu^T H^{a,(j)}(\xi)\nu,\, \xi \in \mathcal{R}^z,\, \nu \in \mathcal{R}^z_\Delta\Big\}. \quad (11)\end{aligned}$$

Note that F is invertible because of the index-1 property, so that we can reformulate (11) to

$$\begin{aligned}&\Delta y(t) \in -F^{-1}\Big(g(z^*) + D\Delta x(t) + E\Delta u(t)\Big) \quad (12)\\ &\oplus \Big\{-\frac{1}{2}F^{-1}\phi \Big| \phi_j = \nu^T H^{a,(j)}(\xi)\nu,\, \xi \in \mathcal{R}^z,\, \nu \in \mathcal{R}^z_\Delta\Big\}.\end{aligned}$$

Inserting (12) into (10) results in a differential inclusion

$$\begin{aligned}\dot{x} \in& f(z^*) + A\Delta x(t) + B\Delta u(t) \\ &- CF^{-1}\big(g(z^*) + D\Delta x(t) + E\Delta u(t)\big) \oplus \mathcal{L} \quad (13)\\ =&(w + \tilde{A}\Delta x(t) + \tilde{B}\Delta u(t)) \oplus \mathcal{L},\end{aligned}$$

where

$$\begin{aligned}w :=& f(z^*) - CF^{-1}g(z^*),\\ \tilde{A} :=& A - CF^{-1}D, \quad (14)\\ \tilde{B} :=& B - CF^{-1}E.\end{aligned}$$

and

$$\begin{aligned}\mathcal{L} =& \Big\{\frac{1}{2}(\sigma - CF^{-1}\phi)\Big|\sigma_i = \nu^T H^{d,(i)}(\xi)\nu, \\ &\phi_j = \nu^T H^{a,(j)}(\xi)\nu,\, \xi \in \mathcal{R}^z,\, \nu \in \mathcal{R}^z_\Delta\Big\}. \quad (15)\end{aligned}$$

We can further simplify (13) by combining the singleton w and the sets $\tilde{B}(\mathcal{U} \oplus (-u^*))$, $\mathcal{L}$ to a new set $\tilde{\mathcal{U}}$:

$$\begin{aligned}&\dot{\tilde{x}} \in \tilde{A}\tilde{x}(t) \oplus \tilde{\mathcal{U}}, \quad (16)\\ &\tilde{x}(t) := \Delta x(t), \quad \tilde{\mathcal{U}} := w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \mathcal{L}.\end{aligned}$$

Note that the set of possible solutions of the differential inclusion (16) is the same as in (13) since uncertain inputs can be considered by enlarging the set of the right-hand side of the differential inclusion (see e.g. [13]).

B. Linearization Error Handling

The problem for evaluating (16) is that the set of linearization errors $\mathcal{L}$ is not known in advance, consequently $\tilde{\mathcal{U}}$ is unknown, too. As an initial guess we enlarge the most recently computed linearization error $\tilde{\mathcal{L}}$ by a user-defined scalar factor $\lambda \in \mathbb{R}^+$, so that

$$\overline{\mathcal{L}} = \hat{c} \oplus \lambda(\tilde{\mathcal{L}} \oplus (-\hat{c})) \quad (17)$$

where $\hat{c}$ is the volumetric center of $\tilde{\mathcal{L}}$. If it turns out that the enclosure assumption ($\overline{\mathcal{L}} \supseteq \mathcal{L}$) is not correct, $\overline{\mathcal{L}}$ has to be further enlarged.

Proposition 1 (Conservativeness of the Abstraction) *If $\overline{\mathcal{L}} \supseteq \mathcal{L}$, the solution of the original dynamics $\gamma(t, x(0), y(0), u(\cdot))$ is enclosed by the solution of the abstracting differential inclusion $\dot{\tilde{x}} \in \tilde{A}\tilde{x}(t) \oplus w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \overline{\mathcal{L}}$.*

Proof: From (16) and $\overline{\mathcal{L}} \supseteq \mathcal{L}$ one obtains a strict model inclusion from the original dynamics to the linear inclusions using the linearization error sets:

$$\begin{aligned}&\forall t \in \tau_k, x \in\ \mathcal{R}^d(\tau_k), y \in \mathcal{R}^a(\tau_k), u \in \mathcal{U}:\\ \dot{x} = f(x,y,u) \in& \tilde{A}(x - x^*) \oplus w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \mathcal{L}\\ \subseteq& \tilde{A}(x - x^*) \oplus w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \overline{\mathcal{L}},\end{aligned}$$

i.e., all solutions of the original dynamics are included by the abstraction using the linearization error $\overline{\mathcal{L}}$. The result is independent of the amount of overapproximation of the reachable sets $\mathcal{R}^d(\tau_k)$, $\mathcal{R}^a(\tau_k)$ and of the amount of overapproximation of $\mathcal{L}$ including the set of possible linearization errors. ■

Clearly, if $\mathcal{L}$ is largely overapproximated, one needs a larger assumption for $\overline{\mathcal{L}}$ and convergence is not guaranteed. In order to ensure convergence, one additionally checks if

$$\overline{\mathcal{L}} \subseteq \mathcal{L}_{\max}, \quad (18)$$

where $\mathcal{L}_{\max}$ is set by the user. If the above inclusion is not fulfilled, the reachable set has to be split in order to reduce the linearization error until (18) is fulfilled. Another possibility is to reduce the time increment r. It is part of future work to find criteria for deciding when it is better to split the reachable sets and when it is better to reduce the time increment r.

In the previous work [6], the linearization error set is guessed as $\overline{\mathcal{L}} = \mathcal{L}_{\max}$, which is constant over all iterations. The time-varying adaption of $\overline{\mathcal{L}}$ in this work significantly reduces the overapproximation by using the previous linearization error and considering the fact that the linearization error changes over time.

The computation of the set of linearization errors is addressed in the next section.

V. Computation of the Linearization Error

The set of linearization errors $\mathcal{L}$ is computed based on the reachable set $\mathcal{R}^d(\tau_k)$ of the linear differential inclusion (16) for the conservative uncertain input

$$\overline{\mathcal{U}} = w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \overline{\mathcal{L}} \supseteq w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \mathcal{L} = \tilde{\mathcal{U}}.$$

In order to overapproximate the set of linearization errors, we first have to reconstruct the reachable set for all variables $\mathcal{R}(\tau_k)$ from the reachable set for the differential variables $\mathcal{R}^d(\tau_k)$.

A. Reachable Set of Differential and Algebraic Variables

We compute the overapproximation of the reachable set for the algebraic variables by replacing Δx with $\mathcal{R}^d_\Delta(\tau_k) := \mathcal{R}^d(\tau_k) \oplus (-x^*)$, Δu with $\mathcal{U}_\Delta := \mathcal{U} \oplus (-u^*)$ in (12), and translating the set by y^*:

$$\mathcal{R}^a(\tau_k) = y^* \oplus (-F^{-1})\big(g(z^*) \oplus D\mathcal{R}^d_\Delta(\tau_k) \oplus E\mathcal{U}_\Delta \oplus \overline{\mathcal{L}}^a(\tau_k)\big). \tag{19}$$

When combining the algebraic and the differential reachable sets, it is important to consider the correlation between both sets, which is evident due to the use of $\mathcal{R}^d_\Delta(\tau_k)$ in (19). For a concise notation we introduce the matrix of generators $G := \begin{bmatrix} g^{(1)} & \dots g^{(p)} \end{bmatrix}$ and the alternative short form of a zonotope $\mathcal{Z}$ as $\mathcal{Z} = (c, G)$.

Proposition 2 (Differential-Algebraic Reachable Set) *Suppose $\mathcal{R}^d(\tau_k) = (c^d, G^d)$, $\mathcal{U} = (c^u, G^u)$, and $\overline{\mathcal{L}}^a = (c^l, G^l)$. An overapproximation for the complete reachable set for both the differential and algebraic variables is*

$$\mathcal{R}(\tau_k) = \left(\begin{bmatrix} c^d \\ c^a \end{bmatrix}, \begin{bmatrix} G^d & \mathbf{0} & \mathbf{0} \\ -F^{-1}DG^d & -F^{-1}EG^u & -F^{-1}G^l \end{bmatrix} \right),$$

where $c^a = y^* - F^{-1}\big(g(z^*) + D(c^d - x^*) + E(c^u - u^*) + c^l\big)$, and $\mathbf{0}$ is a matrix of zeros of proper dimension.

Proof: Using (12), the state of the differential-algebraic system is bounded by

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \begin{bmatrix} x^* \\ y^* - F^{-1}g(z^*) \end{bmatrix} \oplus \begin{bmatrix} I \\ -F^{-1}D \end{bmatrix} \Delta x(t) \oplus \begin{bmatrix} \mathbf{0} \\ -F^{-1}E \end{bmatrix} \Delta u(t) \oplus \begin{bmatrix} \mathbf{0} \\ -F^{-1} \end{bmatrix} \overline{\mathcal{L}}^a.$$

Inserting $\Delta x(\tau_k) \in (c^d - x^*, G^d)$, $\Delta u(\tau_k) \in (c^u - u^*, G^u)$, $\overline{\mathcal{L}}^a(\tau_k) = (c^l, G^l)$ into the above equation yields the proposed computation of $\mathcal{R}(\tau_k)$ using the addition and multiplication rule of zonotopes in (5). ■

Note that Proposition 2 is tighter than the Cartesian product $\mathcal{R}^d(\tau_k) \times \mathcal{R}^a(\tau_k)$ because the latter has as many more generators, as the number of generators of $\mathcal{R}^d(\tau_k)$. Next, $\mathcal{R}(\tau_k)$ is used to overapproximate the set of linearization errors.

B. Bounding the Lagrange Remainder

Using $\mathcal{R}^z = \mathcal{R}(\tau_k) \times \mathcal{U}$ and $\mathcal{R}^z_\Delta = \mathcal{R}^z \oplus (-z^*)$, we first show the computation of the linearization error

$$\mathcal{L}^d \subseteq \frac{1}{2}\Big\{\sigma \Big| \sigma_i = \nu^T H^{d,(i)}(\xi)\nu,\, \xi \in \mathcal{R}^z,\, \nu \in \mathcal{R}^z_\Delta \Big\}, \tag{20}$$

as shown in (10) and then generalize to $\mathcal{L}$. Thereto, we first compute the possible values of the second derivative $\mathcal{H}^{d,(i)} := \{H^{d,(i)}(\xi) | \xi \in \mathcal{R}^z\}$. This is done by first computing the enclosing box $\mathcal{I} := \mathrm{box}(\mathcal{R}^z)$, which is obtained using (7). By applying interval arithmetic, each element of the matrices $H^{d,(i)}(\xi)$ is evaluated for $\xi \in \mathcal{I}$ using interval arithmetic [31].

Interval arithmetic can handle any standard expression, but the result may be rather conservative since dependencies between variables are neglected. To illustrate the effect of dependencies between terms, we first introduce the addition and multiplication rule for the scalar intervals $\mathscr{a} = [\underline{a}, \overline{a}]$ and $\mathscr{b} = [\underline{b}, \overline{b}]$:

$$\begin{aligned} \mathscr{a} \oplus \mathscr{b} =& [\underline{a} + \underline{b}, \overline{a} + \overline{b}], \\ \mathscr{a} \otimes \mathscr{b} =& [\min(\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b}), \max(\underline{a}\,\underline{b}, \underline{a}\,\overline{b}, \overline{a}\,\underline{b}, \overline{a}\,\overline{b})]. \end{aligned} \tag{21}$$

The neglected dependency is best explained by a simple example, where we compute $\mathscr{c} = \mathscr{a}\mathscr{b} \oplus \mathscr{a}$. For $\mathscr{a} = [-2, -1]$ and $\mathscr{b} = [-1, 1]$, we obtain two different results depending on the computation method: $\mathscr{c} = \mathscr{a}\mathscr{b} \oplus \mathscr{a} = [-4, 1]$ and $\tilde{\mathscr{c}} = \mathscr{a}(\mathscr{b} \oplus 1) = [-4, 0]$, where only the latter result is exact. This is because the exact result is obtained from a so-called *single-use expression* in which each interval occurs only once. In the other case, $\mathscr{a}$ appears twice and can take different values when obtaining the lower and upper limits of each operation, although each variable is only allowed to have a single value for each evaluation of the complete expression. This is referred to as the *dependency problem*, which translates to general sets:

$$\begin{aligned} &\underbrace{\{a(b+c) | a \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C}\}}_{\mathcal{A} \otimes (\mathcal{B} \oplus \mathcal{C})} \\ \subseteq &\underbrace{\{ab | a \in \mathcal{A}, b \in \mathcal{B}\} \oplus \{ac | a \in \mathcal{A}, c \in \mathcal{C}\}}_{(\mathcal{A} \otimes \mathcal{B}) \oplus (\mathcal{A} \otimes \mathcal{C})}. \end{aligned} \tag{22}$$

We present a new technique to compute the set of linearization errors, which suffers much less from the dependency problem as in the previous work [6] by first overapproximating (20) with

$$\mathcal{L}^d \subseteq \frac{1}{2}\Big\{\sigma \Big| \sigma_i = \nu^T \mathcal{H}^{d,(i)}\nu,\, \nu \in \mathcal{R}^z_\Delta \Big\}. \tag{23}$$

The new approach uses a newly developed overapproximation of a quadratic map:

Lemma 1 (Quadratic Map) *Given a zonotope $\mathcal{Z} = (c, g^{(1)}, \dots, g^{(p)})$ and a discrete set of matrices $Q^{(i)} \in \mathbb{R}^{n \times n}$, $i = 1 \dots n$, the set*

$$\mathcal{Z}_Q = \{\varphi | \varphi_i = x^T Q^{(i)} x,\, x \in \mathcal{Z}\}$$

is overapproximated by a zonotope

$$\mathrm{quad}(Q, \mathcal{Z}) := (d, h^{(1)}, \dots, h^{(\sigma)})$$

with $\sigma = \binom{p+2}{2} - 1$ generators, where the center is

$$d_i = c^T Q^{(i)} c + 0.5 \sum_{s=1}^{p} g^{(s)^T} Q^{(i)} g^{(s)},$$

and the generators are computed as

$$\begin{aligned} j =&1\dots p: & h_i^{(j)} =& c^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} c \\ j =&1\dots p: & h_i^{(p+j)} =& 0.5 {g^{(j)}}^T Q^{(i)} g^{(j)} \\ l =&\sum_{j=1}^{p-1}\sum_{k=j+1}^{p} 1: & h_i^{(2p+l)} =& {g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)} \end{aligned}$$

The complexity of constructing this zonotope overapproximation with respect to the dimension n is $\mathcal{O}(n^5)$.

Proof: Inserting the definition of a zonotope into the set $\mathcal{Z}_Q = \{\varphi | \varphi_i = x^T Q^{(i)} x,\, x \in \mathcal{Z}\}$ yields

$$\Big\{\varphi \Big| \varphi_i = (c + \sum_{j=1}^{p} \beta_j g^{(j)})^T Q^{(i)} (c + \sum_{j=1}^{p} \beta_j g^{(j)}),\ \beta_j \in [-1,1]\Big\},$$

which can be rearranged to

$$\begin{aligned} \mathcal{Z}_Q = \Big\{\varphi \Big| \varphi_i = & \underbrace{c^T Q^{(i)} c + \sum_{j=1}^{p} 0.5 {g^{(j)}}^T Q^{(i)} g^{(j)}}_{d_i} \\ & + \sum_{j=1}^{p} \beta_j \underbrace{(c^T Q^{(i)} g^{(j)} + {g^{(j)}}^T Q^{(i)} c)}_{h_i^{(j)}} \\ & + \sum_{j=1}^{p} (2\beta_j^2 - 1) \underbrace{0.5 {g^{(j)}}^T Q^{(i)} g^{(j)}}_{h_i^{(p+j)}} \\ & + \sum_{j=1}^{p-1}\sum_{k=j+1}^{p} \beta_j \beta_k \underbrace{({g^{(j)}}^T Q^{(i)} g^{(k)} + {g^{(k)}}^T Q^{(i)} g^{(j)})}_{h_i^{(2p+l)}}, \\ & \beta_i \in [-1,1]\Big\} \subseteq \Big(d, h^{(1)}, \dots, h^{(\sigma)}\Big). \end{aligned}$$

The obtained zonotope is an overapproximation since $\beta_j \in [-1,1]$, $(2\beta_j^2 - 1) \in [-1,1]$, and $\beta_j \beta_k \in [-1,1]$ for $j \neq k$. The number of new generators is obtained from the fact that the new generators $h^{(j)}$ are computed by picking two elements from the set containing all generators and the center, where replacement is allowed and order does not matter. By subtracting the possibility that one can choose two centers, one obtains $\sigma = \binom{p+2}{2} - 1$ generators.

It remains to derive the complexity. Quadratic operations such as ${g^{(j)}}^T Q^{(i)} g^{(k)}$ have complexity $\mathcal{O}(n^2)$. The number p of generators of $\mathcal{Z}$ can be expressed by its order as $\rho\, n$, such that the resulting zonotope has $\binom{(\rho\, n)+2}{2} - 1$ generators, a number which can be bounded by $\mathcal{O}(n^2)$, such that we have $\mathcal{O}(n^4)$ for all generator computations for each dimension and $\mathcal{O}(n^5)$ for all dimensions. ■

The above Lemma is used in the proof of the following Proposition to overapproximate $\mathcal{L}^d$ in (23).

Proposition 3 (Linearization Error) *Let each $\mathcal{H}^{d,(i)}$ be bounded by an interval matrix, the linearization error according to* (23) *is overapproximated by first computing* $\mathrm{quad}(\mathcal{H}^d, \mathcal{R}^z_\Delta)$ *according to Lemma 1, except that the center and generators are computed via interval arithmetic, so that one obtains an interval center $\mathscr{d} = d_c \oplus [-d_\Delta, d_\Delta]$, $d_c, d_\Delta \in \mathbb{R}^n$ and interval generators $\mathscr{h}^{(i)} = h_c^{(i)} \oplus [-h_\Delta^{(i)}, h_\Delta^{(i)}]$, $h_c, h_\Delta \in \mathbb{R}^n$. Using these interval-valued results, the overapproximating zonotope with real-valued center and generators is*

$$\mathcal{L}^d \subseteq \frac{1}{2}\Big\{z^T \mathcal{H}^{d,(i)} z \Big| z \in \mathcal{R}^z_\Delta\Big\} \subseteq \frac{1}{2}\mathrm{quad}^{\mathrm{int}}(\mathcal{H}^d, \mathcal{R}^z_\Delta),$$

where

$$\mathrm{quad}^{\mathrm{int}}(\mathcal{H}^d, \mathcal{R}^z_\Delta) := (d_c, h_c^{(1)}, \dots, h_c^{(\sigma)}, l^{(1)}, \dots, l^{(n)}),$$

and

$$l_j^{(m)} = \begin{cases} d_{\Delta,j} + \sum_{i=1}^{\sigma} h_{\Delta,j}^{(i)},\ \textit{for } m = j \\ 0,\ \textit{for } m \neq j \end{cases}.$$

The complexity with respect to the dimension n is $\mathcal{O}(n^5)$.

Proof: The interval valued center $\mathscr{d}$ and generators $\mathscr{h}^{(i)}$ represent the set

$$\begin{aligned} & \underbrace{(d_c \oplus [-d_\Delta, d_\Delta])}_{=\mathscr{d}} \bigoplus_{i=1}^{\sigma} \Big([-1,1] \otimes \underbrace{(h_c^{(i)} \oplus [-h_\Delta^{(i)}, h_\Delta^{(i)}]}_{=\mathscr{h}^{(i)}}\Big) \\ & \overset{(22)}{\subseteq} \underbrace{d_c \bigoplus_{i=1}^{\sigma} \Big([-1,1] \otimes h_c^{(i)}\Big)}_{=(d_c, h_c^{(1)}, \dots, h_c^{(\sigma)})} \\ & \underbrace{\oplus [-d_\Delta, d_\Delta] \bigoplus_{i=1}^{\sigma} \Big([-1,1] \otimes [-h_\Delta^{(i)}, h_\Delta^{(i)}]\Big)}_{=[-y,y], \quad y = d_\Delta + \sum_{i=1}^{\sigma} h_\Delta^{(i)}}. \end{aligned}$$

It remains to reformulate the multidimensional interval $[-y, y]$ to the zonotope $(\mathbf{0}, l^{(1)}, \dots, l^{(n)})$, where $\mathbf{0}$ is a vector of zeros of proper dimension and $l^{(i)}$ are the generators from the proposition. ■

The above technique uses interval arithmetic extensively, which slows down the computation compared to Lemma 1. The following lemma shows that interval arithmetic can be avoided for multiplication with symmetric intervals.

Lemma 2 (Symmetric Interval Matrix Multiplication) *[2, Lemma 3.2] For $N \in \mathbb{R}^{m \times q}$ and an interval matrix $\mathcal{S} = [-S, S]$ with symmetric bound $S \in \mathbb{R}^{q \times l}$,*

$$N\mathcal{S} = \big[-|N|S, |N|S\big], \quad \mathcal{S}^T N^T = \big[-S^T|N^T|, S^T|N^T|\big],$$

where the absolute value is applied elementwise.

Inspired by this result, we use an alternative computation of Proposition 3, which surprisingly returns the exact same result without interval arithmetic.

Theorem 1 (Interval-Free Computation of $\mathcal{L}^d$) *After introducing $H_c^d, H_\Delta^d \in \mathbb{R}^{n \times n}$ such that $\mathcal{H}^d = H_c^d \oplus [-H_\Delta^d, H_\Delta^d]$, the equality*

$$\begin{gathered} \mathrm{quad}^{\mathrm{int}}(\mathcal{H}^d, \mathcal{R}^z_\Delta) = \mathrm{quad}(H_c^d, \mathcal{R}^z_\Delta) \oplus [-\eta, \eta], \\ \eta := |\mathcal{R}^z_\Delta|^T H_\Delta^d |\mathcal{R}^z_\Delta| \end{gathered}$$

holds.

Proof: For simplicity of notation, we first introduce the vectors $\lambda^{(i)}$, where $\lambda^{(1)} = c$ represents the center, and

$\lambda^{(2)}, \dots, \lambda^{(\sigma+1)}$ the generators of $\mathcal{R}^z_\Delta$. In Lemma 1, the operation $\mathrm{quad}(\mathcal{H}^d, \mathcal{R}^z_\Delta)$ is broken down into expressions of the form

$$\begin{gathered}{\lambda^{(i)}}^T (H^d_c \oplus [-H^d_\Delta, H^d_\Delta])\lambda^{(j)} = \\ \underbrace{{\lambda^{(i)}}^T H^d_c \lambda^{(j)}}_{\text{fixed value}} \oplus \underbrace{{\lambda^{(i)}}^T [-H^d_\Delta, H^d_\Delta]\lambda^{(j)}}_{\text{symmetric value}},\end{gathered}$$

where the equality follows from the fact that the set-valued components have only a single occurrence. Due to the equality and the fact that each result has a fixed and symmetric part, we can conclude that the values d_c and $h_c^{(m)}$ in Proposition 3 exactly match ${\lambda^{(i)}}^T H^d_c \lambda^{(j)}$ for corresponding choices of i, j. Thus, using the variables from Proposition 3, we have that

$$\begin{aligned}\mathrm{quad}(H^d_c, \mathcal{R}^z_\Delta) =& (d_c, h_c^{(1)}, \dots, h_c^{(\sigma)}) \\ \mathrm{quad}([-H^d_\Delta, H^d_\Delta], \mathcal{R}^z_\Delta) =& ([-d_\Delta, d_\Delta], [-h_\Delta^{(1)}, h_\Delta^{(1)}], \dots, \\ & [-h_\Delta^{(\sigma)}, h_\Delta^{(\sigma)}]) \\ =& (\mathbf{0}, l^{(1)}, \dots, l^{(n)}).\end{aligned}$$

Using the above result, one can conclude that

$$\begin{gathered}\mathrm{quad}^{\mathrm{int}}(\mathcal{H}^d, \mathcal{R}^z_\Delta) = (d_c, h_c^{(1)}, \dots, h_c^{(\sigma)}, l^{(1)}, \dots, l^{(n)}) = \\ \mathrm{quad}(H^d_c, \mathcal{R}^z_\Delta) \oplus \mathrm{quad}([-H^d_\Delta, H^d_\Delta], \mathcal{R}^z_\Delta)\end{gathered}$$

One can further simplify $\mathrm{quad}([-H^d_\Delta, H^d_\Delta], \mathcal{R}^z_\Delta)$ to

$$\begin{gathered}\mathrm{quad}([-H^d_\Delta, H^d_\Delta], \mathcal{R}^z_\Delta) = \\ \bigoplus_{i=1}^{p+1} \bigoplus_{j=1}^{p+1} \Big([-1,1] \lambda^{(i)} [-H^d_\Delta, H^d_\Delta] \lambda^{(j)} \Big) \overset{Lemma\,2}{=} \\ [-1,1] \sum_{i=1}^{p+1} \sum_{j=1}^{p+1} \Big(|\lambda^{(i)}| H^d_\Delta |\lambda^{(j)}| \Big) = \\ [-1,1] \underbrace{\Big(\sum_{i=1}^{p+1} |\lambda^{(i)}| \Big)}_{=|\mathcal{R}^z_\Delta|, \text{ see (7)}} H^d_\Delta \underbrace{\Big(\sum_{j=1}^{p+1} |\lambda^{(j)}| \Big)}_{=|\mathcal{R}^z_\Delta|, \text{ see (7)}}\end{gathered}$$

which concludes the proof. ■

It follows from Lemma 1 that the linearization error computation using Theorem 1 has complexity $\mathcal{O}(n^5)$ with respect to the dimension n.

In the previous work [6], the set of linearization errors is bounded using $\mathcal{I}_\Delta := \mathrm{box}(\mathcal{R}^z_\Delta)$ by

$$\begin{gathered}\mathcal{L}^d_i \subseteq \mathcal{I}^T_\Delta \otimes \mathcal{H}^{d,(i)} \otimes \mathcal{I}_\Delta \subseteq [-\overline{L}_i, \overline{L}_i], \\ \overline{L}_i = |\mathcal{I}_\Delta|^T \, |\mathcal{H}^{d,(i)}| \, |\mathcal{I}_\Delta|.\end{gathered} \tag{24}$$

The computational complexity of (24) with respect to the system dimension is $\mathcal{O}(n^3)$ since the quadratic evaluation for the i^{th} coordinate is $\mathcal{O}(n^2)$ and there are n coordinates. Depending on the nonlinear dynamics, the linearization error computation in (24) might suffer from substantial overapproximation. For the power system example in Sec. VII, one cannot even compute the first time interval using the technique from [6] since the linearization error computation does not stabilize.

It remains to compute the linearization error $\mathcal{L}$ in (15) using the techniques presented above.

Corollary 1 (Interval-Free Computation of $\mathcal{L}$) *Given the zonotopes* $\mathcal{L}^d_c = \mathrm{quad}(H^d_c, \mathcal{R}^z_\Delta) = (d, H)$ *and* $\mathcal{L}^a_c = \mathrm{quad}(H^a_c, \mathcal{R}^z_\Delta) = (e, V)$ *computed as presented in Lemma 1, the overapproximative set of linearization errors is computed as*

$$\begin{gathered}\mathcal{L} = \underbrace{\frac{1}{2}(d - CF^{-1}e, H - CF^{-1}V)}_{=:\mathcal{L}_c} \\ \oplus \underbrace{\frac{1}{2}\big([-\zeta, \zeta] \oplus (-CF^{-1})[-\varrho, \varrho]\big)}_{=:\mathcal{L}_\Delta}, \\ \zeta := |\mathcal{R}^z_\Delta|^T H^d_\Delta |\mathcal{R}^z_\Delta|, \quad \varrho := |\mathcal{R}^z_\Delta|^T H^a_\Delta |\mathcal{R}^z_\Delta|.\end{gathered}$$

Proof: We split the computation of the Lagrangian remainder in (15) as follows:

$$\begin{aligned}\mathcal{L} = & \Big\{ \frac{1}{2}(\sigma - CF^{-1}\phi) \Big| \sigma_i = \nu^T H^{d,(i)}(\xi)\nu, \\ & \quad \phi_j = \nu^T H^{a,(j)}(\xi)\nu, \, \xi \in \mathcal{R}^z, \, \nu \in \mathcal{R}^z_\Delta \Big\} \\ \subseteq & \Big\{ \frac{1}{2}(\sigma - CF^{-1}\phi) \Big| \sigma_i = \nu^T H^{d,(i)}_c \nu, \, \phi_j = \nu^T H^{a,(j)}_c \nu, \, \nu \in \mathcal{R}^z_\Delta \Big\} \\ & \oplus \Big\{ \frac{1}{2}(\sigma - CF^{-1}\phi) \Big| \sigma_i = \nu^T [-H^{d,(i)}_\Delta, H^{d,(i)}_\Delta]\nu, \\ & \quad \phi_j = \nu^T [-H^{a,(j)}_\Delta, H^{a,(j)}_\Delta]\nu, \, \nu \in \mathcal{R}^z_\Delta \Big\} \\ \overset{\text{Theorem 1}}{\subseteq} & \; \mathcal{L}_c \oplus \mathcal{L}_\Delta\end{aligned}$$

We overapproximate $\mathcal{L}_c$ by using the fact that $\mathrm{quad}(H^d_c, \mathcal{R}^z_\Delta)$ and $\mathrm{quad}(H^a_c, \mathcal{R}^z_\Delta)$ return zonotopes with generators multiplied by the same sequence of multipliers β_i and β_j, see Lemma 1. From this follows that $\mathcal{L}_c \subseteq (d + CF^{-1}e, H + CF^{-1}V)$, where it is additionally considered that the multiplication with CF^{-1} is computed such that the sequence of scalar multipliers is preserved. ■

The overapproximative computation of the linearization error as well as the other presented techniques are assembled in the next section to the overall algorithm of the proposed approach.

VI. Overall Algorithm

The overall algorithm for computing an overapproximation of the reachable set for the nonlinear, semi-explicit, index-1 DAE in (1) is summarized in Alg. 1. The overapproximation of the reachable set is obtained by combining the conservative linearization procedure, the previously known techniques for reachability analysis of linear differential inclusions, and the improved approach for computing the linearization error. The algorithm consists of three major parts:

① Computing a linearization and the corresponding set of linearization errors $\mathcal{L}$ of the current time interval. The overapproximated reachable set $\mathcal{R}(\tau_s)$ based on the assumption of linearization errors $\overline{\mathcal{L}}$ is obtained as a by-product from this computation.

② Splitting of reachable sets when the set of linearization errors $\mathcal{L}$ is too large.

③ Computing the tightly overapproximated reachable set at the next point in time $\mathcal{R}(t_{s+1})$ using the set of linearization errors $\mathcal{L}$. It is crucial that $\mathcal{R}(t_{s+1})$ is tightly overapproximated since the reachable set of the next point in time and the next time interval are based on this set. This is in contrast to the reachable set for the time interval, since it only has a small contribution on the wrapping-effect by influencing the size of $\mathcal{L}$.

These steps are also reflected in Alg. 1. The linearization error computation in ① starts by computing the coefficients A, B, C, D, E, F in (9) for the linearization point z^* of the current time interval, which are combined to w, $\tilde{A}$, $\tilde{B}$ according to (14), a process which is abbreviated by $\texttt{taylor}(\mathcal{R}^d(t_k)) \rightarrow w, \tilde{A}, \tilde{B}$ in line 3 of Alg. 1. The assumption on the linearization error is computed in line 5 by slightly enlarging the set of the previous linearization errors. In line 7, the reachable set $\mathcal{R}^d(\tau_k)$ for a whole time interval is computed based on the linear differential inclusion in (16) and the uncertain input $\overline{\mathcal{U}}$ (line 6), which in turn is based on $\overline{\mathcal{L}}$ (line 5). The reachable set of the differential variables is complemented to the set of all variables in line 8, which is then used in line 9 to overapproximate the set of linearization errors. If $\mathcal{L} \subseteq \overline{\mathcal{L}}$, $\mathcal{R}^d(\tau_k)$ is a valid overapproximation, see Proposition 1, otherwise, the loop has to be repeated for a new assumption on the linearization error.

When the reachable set is split into $\mathcal{R}^{(1)}(t_s), \mathcal{R}^{(2)}(t_s)$ in line 12, which is indicated by $\texttt{split}(\mathcal{R}(t_k)) \rightarrow \mathcal{R}^{(1)}(t_s), \mathcal{R}^{(2)}(t_s)$, one has to recursively call Alg. 1 for the remaining time horizon.

Finally, the reachable set for the next point in time is computed in line 19 using the linearization error $\mathcal{L}$ so that one obtains the uncertain input $\tilde{\mathcal{U}} \subseteq \overline{\mathcal{U}}$ for the linear differential inclusion in (16). The reachable set for the time interval is not re-computed using the refined input $\tilde{\mathcal{U}}$ since its contribution to the wrapping-effect is marginal as previously discussed.

VII. Power System Example

We apply our approach to a power system problem. The verification task is to show that after a power drop-out of a power plant and its subsequent reconnection to the grid, the system state comes back to its original operating point. We show this for a set of initial states by computing the reachable set of the differential variables until it is enclosed by the initial set again. This problem is known as *transient stability analysis* in the power system literature [34]. The power system is modeled by the IEEE 14-bus benchmark power system network to which we add power generators, which provide the dynamic part, while the algebraic part originates from the power system network. The model has 14 differential and 28 algebraic variables, giving a total of 42 continuous state variables.

Note that guaranteed transient stability analysis cannot be achieved using Monte-Carlo simulation since the set of initial states is uncertain, so it is possible to miss the simulations which do not return to the original operating point. Other methods for analyzing transient stability besides Monte Carlo simulation are summarized in [39]. Besides numerical simulation, the other model-based techniques are based on Lyapunov methods for determining regions of attraction, i.e. regions from where all trajectories converge to the operating point of the power system [1], [15], [45]. Lyapunov methods are a formal technique, but they suffer from (1) conservative results, meaning that the region of attraction is often largely under-approximated for larger systems, and (2) require simplified models. Especially in the multi-machine case, one often has to neglect transfer conductances and replace some time-varying generator states by constants. There also exists a number of model-free analysis techniques, such as pattern recognition, expert systems, and neural nets, see [39].

Algorithm 1 $\texttt{reach}(\mathcal{R}(0), t_f, r, \mathcal{U}, \mathcal{L}^{\max}, \lambda)$

```
Require: Initial set $\mathcal{R}^d(0)$, input set $\mathcal{U}$, time horizon $t_f$, time
    step $r$, max. linearization error $\mathcal{L}^{\max}$, factor $\lambda$
Ensure: $\mathcal{R}^d([0,t_f])$
1: $t_0 = 0$, $k = 0$, $\mathcal{L} = \{\mathbf{0}\}$, $\mathcal{R}^{union} = \emptyset$, $u^* = \texttt{center}(\mathcal{U})$
2: while $t_k < t_f$ do
3:     $\texttt{taylor}(\mathcal{R}^d(t_k)) \rightarrow w, \tilde{A}, \tilde{B}$ (see (9), (14))
4:     repeat
5:         $\overline{\mathcal{L}} = \hat{c} \oplus \lambda(\mathcal{L} \oplus (-\hat{c}))$ (see (17))
6:         $\overline{\mathcal{U}} = w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \overline{\mathcal{L}}$ (see (16))
           $\mathcal{R}^d(\tau_k) = \mathrm{CH}\big(\mathcal{R}^d(t_k), e^{\tilde{A}r}\mathcal{R}^d(t_k) \oplus \Gamma(r)u_c\big)$
7:             $\oplus \mathcal{R}^d_\epsilon \oplus \mathcal{R}^d_p\big(\overline{\mathcal{U}} \oplus (-u_c), r\big)$ (see (4))        ①
8:         compute $\mathcal{R}(\tau_k)$ using Prop. 2
9:         compute $\mathcal{L}$ using Corollary 1
10:    until $\mathcal{L} \subseteq \overline{\mathcal{L}} \vee \mathcal{L} \nsubseteq \mathcal{L}^{\max}$
11:    if $\mathcal{L} \nsubseteq \mathcal{L}^{\max}$ then
12:        $\texttt{split}(\mathcal{R}(t_k)) \rightarrow \mathcal{R}^{(1)}(t_k), \mathcal{R}^{(2)}(t_k)$
13:        $\mathcal{R}^{(1)}([t_k,t_f]) = \texttt{reach}(\mathcal{R}^{(1)}(t_k), (t_f - t_k), ...)$
14:        $\mathcal{R}^{(2)}([t_k,t_f]) = \texttt{reach}(\mathcal{R}^{(2)}(t_k), (t_f - t_k), ...)$
           $\mathcal{R}^{union} = \mathcal{R}^{union} \cup \mathcal{R}^{(1)}([t_k,t_f])$        ②
15:                $\cup \mathcal{R}^{(2)}([t_k,t_f])$
16:        $t_k = t_f$
17:    else
18:        $\tilde{\mathcal{U}} = w \oplus \tilde{B}(\mathcal{U} \oplus (-u^*)) \oplus \mathcal{L}$ (see (16))
           $\mathcal{R}^d(t_{k+1}) = e^{\tilde{A}r}\mathcal{R}^d(t_k) \oplus \Gamma(r)u_c$
19:            $\oplus \mathcal{R}^d_p(\tilde{\mathcal{U}} \oplus (-u_c), r)$ (see (4))        ③
20:        $\mathcal{R}^{union} = \mathcal{R}^{union} \cup \mathcal{R}^d(\tau_k)$
21:        $t_{k+1} = t_k + r$, $k := k + 1$
22:    end if
23: end while
24: $\mathcal{R}^d([0,t_f]) = \mathcal{R}^{union}$
```

Reachability analysis for power systems has been considered in [16], [32], [49], but only for small problems. In [32], transient stability analysis is performed using level-sets for a single-machine-infinite-bus system modeled by ODEs with only 2 state variables. A slightly larger double-machine-infinite-bus system with 2 buses described by ODEs with 5 state variables is considered in [49]. There, the reachability analysis is performed by checking if cells of a discretized state space can be reached from other cells by numerical simulation. In [16], an initial DAE model is simplified to ODEs and further to linear ODEs, without considering errors made during each conversion. A 3-bus system is considered in that work, and effects on wind variability rather than transient stability are investigated.

We first present the mathematical model of the power system and then show the results of the reachability analysis.

A. Mathematical Model

We use the IEEE 14-bus benchmark system enhanced by generator dynamics, which is depicted in Fig. 2. In order to obtain the correct equations for the relatively complex 14-bus system, we auto-generate the equations using symbolic computations in MATLAB. First, the power flow equations are obtained according to [46] for each bus, where variable indices refer to the bus number. The absolute value of the bus voltage is denoted by $|V_i|$ [p.u] (p.u.: per unit), the angle of the bus voltage by Θ_i [rad], the active power by P_i [p.u.], and the reactive power by Q_i [p.u.], where inflow of power is positive. The buses are connected via admittances $Y_{ij} = Y_{ji}$, where i and j are the indices of the connected buses. The absolute value and the angle of the admittances are denoted by $|Y_{ij}|$ and $\Psi_{ij} = \angle Y_{ij}$, respectively. The active and reactive power of each bus results from the generator production $P_{g,i}, Q_{g,i}$ and a demand of that node $P_{d,i}, Q_{d,i}$. In order to compute the generated power, we additionally have to introduce the generator voltage E_i [p.u.], the generator phase angle $\tilde{\delta}_i$ [rad], and the admittance from the generator to the i^{th} generator bus $Y_{g,i}$, where $|Y_{g,i}|$ [p.u.], $\Psi_{g,i} = \angle Y_{g,i}$ [rad] are the absolute values and phase angles, respectively.

We reorder the numbering of the power network buses, where N_g is the number of generators and N_l is the number of load buses. In this work, the first bus ($i = 1$) is connected to a generator and serves as the slack bus, whose generator phase angle $\tilde{\delta}_1$ is the reference for all other generator phase angles $\delta_i := \tilde{\delta}_i - \tilde{\delta}_1$ and bus phase angles $\Theta_i := \tilde{\Theta}_i - \tilde{\delta}_1$. Further, the power system has N_g so-called *generator buses*, which are connected to generators. Those buses (including the slack bus) produce active and reactive power according to the following equations (see [46]):

$$\begin{aligned} P_{g,i} &= E_i V_i |Y_{g,i}| \cos(\Psi_{g,i} + \delta_i - \Theta_i) - V_i^2 |Y_{g,i}| \cos(\Psi_{g,i}), \\ Q_{g,i} &= -E_i V_i |Y_{g,i}| \sin(\Psi_{g,i} + \delta_i - \Theta_i) + V_i^2 |Y_{g,i}| \sin(\Psi_{g,i}). \end{aligned}$$

The remaining N_l buses are referred to as *load buses* ($i = N_g + 1 \dots N_g + N_l$), which are power sinks. The power flow equations as in [46, p.174] of each bus are

$$\begin{aligned} P_i = P_{g,i} + P_{d,i} &= \sum_{j=1}^{N_g+N_l} V_i V_j |Y_{ij}| \cos(\Psi_{ij} + \Theta_j - \Theta_i), \\ Q_i = Q_{g,i} + Q_{d,i} &= -\sum_{j=1}^{N_g+N_l} V_i V_j |Y_{ij}| \sin(\Psi_{ij} + \Theta_j - \Theta_i). \end{aligned} \tag{25}$$

So far, only the algebraic equations of the power system are introduced. The dynamic equations are described by the generator dynamics. For simplicity, we use the same model for all generators and synchronous condensers, where the latter are generators that produce no active power. The variables of the i^{th} generator are the voltage angle δ_i [rad], the angular velocity ω_i [rad/s], and the torque $T_{m,i}$ [p.u.], and the commanded powers $P_{c,i}$ [p.u.]. The following set of differential equations describes the generator dynamics [16]:

$$\begin{aligned} \dot{\delta}_i &= \omega_i - \omega_1 \\ \dot{\omega}_i &= -\frac{D_i}{M_i}(\omega_i - \omega_1) + \frac{1}{M_i} T_{m,i} - \frac{1}{M_i} P_{g,i} \\ \dot{T}_{m,i} &= -\frac{1}{T_{SV,i} R_{D,i} \omega_s}(\omega_i - \omega_s) - \frac{1}{T_{SV,i}} T_{m,i} + \frac{1}{T_{SV,i}} P_{c,i}, \end{aligned} \tag{26}$$

where M_i [MJ/Hz^2] is the rotational inertia, D_i [s/rad] the damping coefficient, $T_{SV,i}$ [s] is the time constant of the governor, and $\frac{1}{R_{D,i}}$ [-] is the proportional gain of the governor. For $i = 1$, the dynamics is solely described by ω and T_m since the phase angle is always 0.

The power drop-out of the i^{th} power plant is modeled by setting the active and reactive power in (25) and (26) to zero ($P_{g,i} = 0$, $Q_{g,i} = 0$). In order to write the power system in the standard form of time-invariant, semi-explicit, index-1 DAEs presented in (1), we rename the dynamic, algebraic, and input variables. The algebraic variables are changed to

$$\begin{aligned} &i = 1 \dots N_g: && y_i = E_i, \\ &i = 1 \dots N_l: && y_{N_g+i} = V_{N_g+i}, \\ &i = 1 \dots (N_g + N_l): && y_{N_g+N_l+i} = \Theta_i, \end{aligned}$$

the dynamic variables to

$$\begin{aligned} &i = 2 \dots N_g: && x_{i-1} = \delta_i, \\ &i = 1 \dots N_g: && x_{N_g+i-1} = \omega_i, \\ &i = 1 \dots N_g: && x_{2N_g+i-1} = T_{m,i}, \end{aligned}$$

and the inputs to

$$i = 1 \dots N_g: \quad u_i = P_{c,i}.$$

When the i^{th} power plant is not on the grid, the variable E_i is removed from (25), (26), and is no longer an unknown variable. We replace $y_i = E_i$ by $y_i = V_i$ during the power drop-out, since the power plant can no longer control the voltage at the i^{th} bus.

The generator parameters of the IEEE 14-bus system are listed in Tab. I and we refer to [51] for the parameters of the power grid.

TABLE I
PARAMETERS OF THE GENERATORS.

$\forall i$:	M_i	D_i	$\lvert Y_{g,i}\rvert$	$\Psi_{g,i}$	$T_{SV,i}$	$R_{D,i}$	ω_s
	$\frac{1}{15\pi}$	0.04	5	$-\frac{\pi}{2}$	1	0.05	120π

B. Reachability Analysis

We investigate the transient stability by a power drop-out of the largest power plant at bus 1. The power system is in normal operation for the first time interval $t = [0, 0.1]$ [s], which we call *pre-fault* phase. In the time interval $t = [0.1, 0.13]$ [s], the power plant at bus 1 producing the most power is taken off the grid, which we refer to as the *fault-on* phase. At $t = 0.13$ [s], the power plant is reconnected, which starts the *post-fault* phase. The reachable set computation is stopped when the reachable set of differential variables is enclosed by the initial set of states, proving that all differential state variables

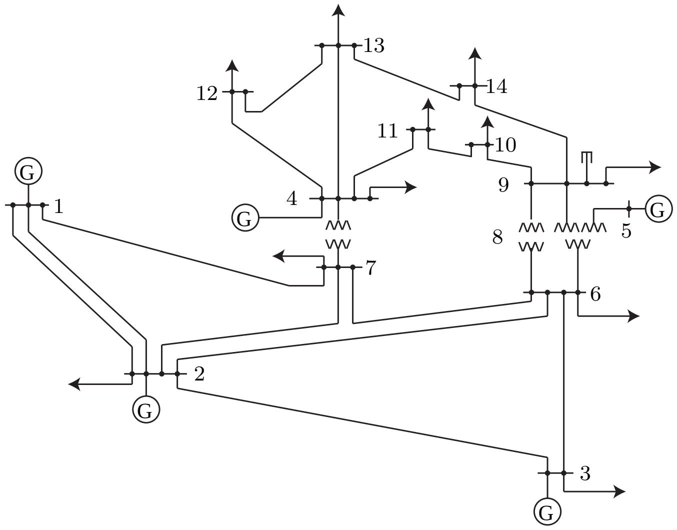


Fig. 2. IEEE 14-bus benchmark system.

return to the original operating point (steady state). We choose the set of initial states as $\forall i : \delta_i(0) \in \delta_i^0 \oplus 0.01 \cdot [-1, 1]$, $\omega_i(0) \in \omega_i^0 \oplus 0.1 \cdot [-1, 1]$, $T_{m,i}(0) \in T_{m,i}^0 \oplus 0.001 \cdot [-1, 1]$, where the superscripted zero refers to the steady state solution.

For the reachability analysis we use a time increment of $r = 0.001$ [s] in the fault-on mode, $r = 0.005$ [s] for the pre-fault mode and the post-fault mode until $t = 2$ [s], and $r = 0.02$ [s] for the remaining time in the post-fault mode. We restrict the order of zonotopes to $\rho = 400$ and the order of zonotopes for the linearization error computation to $\tilde{\rho} = 3$ using the order reduction method in [25].

The reachable sets of the entire time horizon for selected projections onto differential and algebraic variables are shown in Fig. 3. The simulations of system trajectories from randomly chosen initial states are indicated by black lines. Note that the algebraic values jump when the power plant is taken off the grid and when it is reconnected to the grid. At time $t = 4.32$ [s], the initial set is reached after 540 iterations, which is shown for selected projections in Fig. 4. For the entire time horizon, it is not required to split the reachable set.

The computations took 3889 [s] to compute in MATLAB on an i7 Processor and 6GB memory. Around a third of the computation time is spent on computing the set of Hessian matrices $\mathcal{H}^d$ and $\mathcal{H}^a$ using interval arithmetic, and another third on computing the linearization error for a given set of Hessian matrices. Note that the Hessian matrices have to be computed anyway, even when using the linearization error computation of the previous work [6].

We are not able to compare the obtained reachable sets with other methods, since none of the previous work on systems with DAEs would scale to the size of the problem presented here (to the best knowledge of the authors). On a further note, the difficulty of reachability analysis problems not only depends on the number of state variables, but also how well a method fits to the characteristics of the problem, the combination of nonlinear terms in differential equations, and the size of the initial set of states as well as the input set. To the best knowledge of the authors, there is no publication that ranks nonlinear expressions in terms of difficulty for reachability analysis.

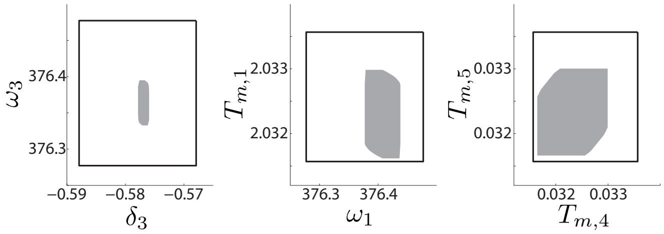


Fig. 4. Reachable set of selected projections at time $t = 4.32$ [s] when all solutions have returned to the initial set. The dark gray area shows the reachable set and the black box shows the initial set.

VIII. Conclusion

This paper presents an approach for overapproximatively computing reachable sets of systems with differential-algebraic equations. The presented method scales favorably with the system dimension due to the use of zonotopes for the set representation. Our approach can be applied to any nonlinear, semi-explicit, index-1 DAE system and any nonlinear system of ODEs with unique solutions for all consistent initial states. It has been shown that the computation of the set of linearization errors has drastically improved compared to the previous method [6], which already cannot be stabilized at the first time interval without splitting. When the set of linearization errors becomes too large, it is necessary to split the reachable set as described in [6]. When many splits are required, it is advantageous to use zonotope bundles as a set representation [3]. The method can be integrated in the reachability analysis of hybrid systems by considering changes in the continuous dynamics when hitting so called *guard sets* [4], [23]. Future work should focus on further investigating on how to compute tighter linearization error bounds.

Acknowledgment

This research was supported in part by U.S. National Science Foundation grant number CCF-0926181 and the U.S. Air Force Office of Scientific Research grant number FA9550-06-1-0312. The authors would also like to thank Milos Cvetkovic for helping to set up the power system example.

References

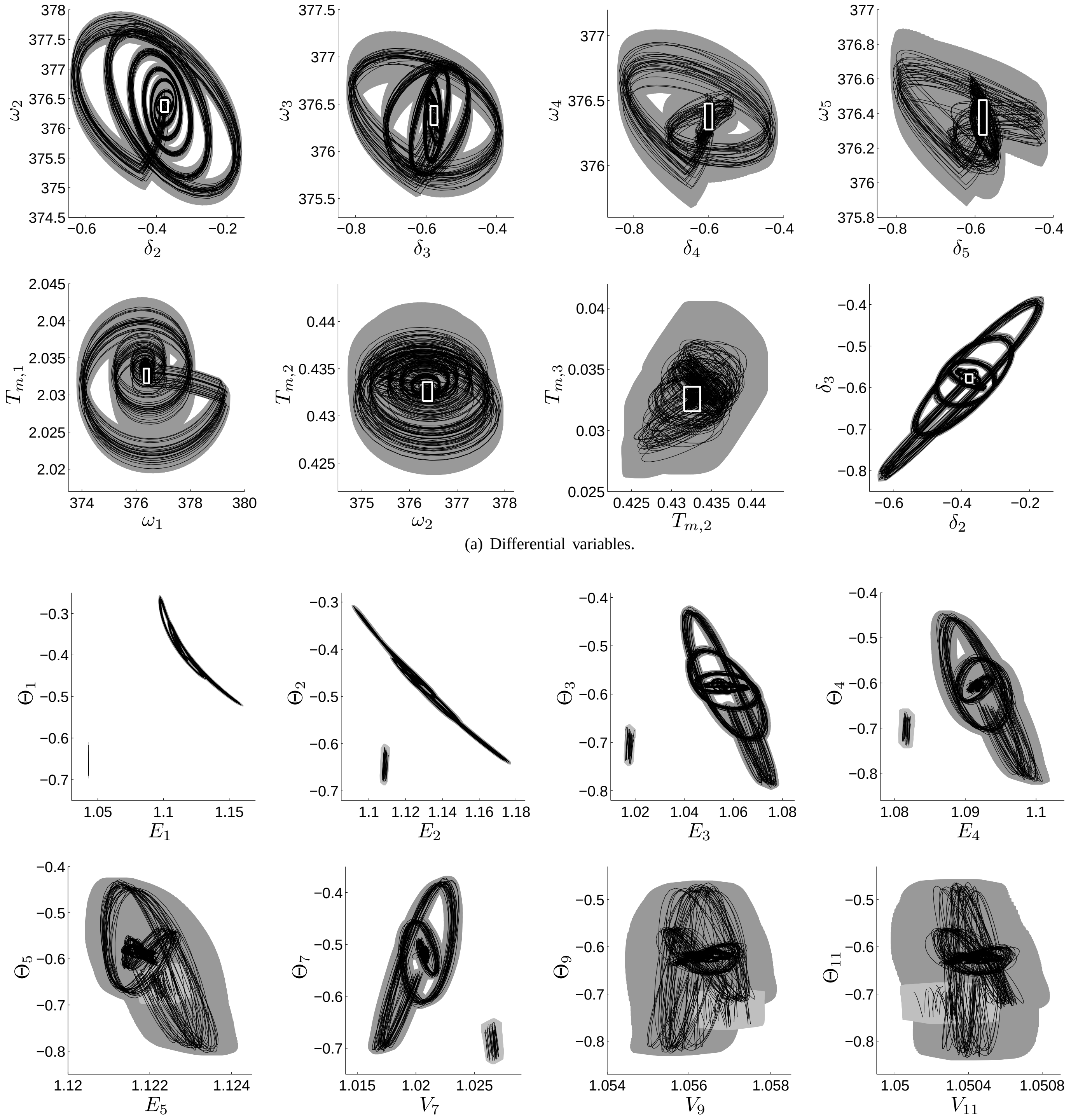
[1] L. F. C. Alberto, F. H. J. R. Silva, and N. G. Bretas. Direct methods for transient stability analysis in power systems: State of art and future perspectives. In *Proc. of the IEEE Porto Power Tech Conference*, 2001.

[2] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. Dissertation, Technische Universität München, 2010. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4.

[3] M. Althoff and B. H. Krogh. Zonotope bundles for the efficient computation of reachable sets. In *Proc. of the 50th IEEE Conference on Decision and Control*, pages 6814–6821, 2011.

[4] M. Althoff and B. H. Krogh. Avoiding geometric intersection operations in reachability analysis of hybrid systems. In *Hybrid Systems: Computation and Control*, pages 45–54, 2012.

[5] M. Althoff, C. Le Guernic, and B. H. Krogh. Reachable set computation for uncertain time-varying linear systems. In *Hybrid Systems: Computation and Control*, pages 93–102, 2011.



(a) Differential variables.

(b) Algebraic variables; the light gray area shows the reachable set during fault-on operation.

Fig. 3. Reachable set of the entire time horizon for selected projections of differential variables. Black lines show random simulations, the gray area shows the reachable set, and the white box the initial set.

[6] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[7] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[8] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, and O. Maler. Recent progress in continuous and hybrid reachability analysis. In *Proc. of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 1582–1587, 2006.

[9] E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Control and Computation*, pages 20–35, 2003.

[10] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM: Society for Industrial and Applied Mathematics, 1998.

[11] J. P. Aubin and A. Cellina. *Differential Inclusions: Set-Valued Maps and Viability Theory*. Springer, 1984.

[12] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4:83–

97, 1998.

[13] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, LNCS 1790, pages 73–88. Springer, 2000.

[14] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*. North-Holland, 1989.

[15] H.-D. Chang, C.-C. Chu, and G. Cauley. Direct stability analysis of electric power systems using energy functions: Theory, applications, and perspective. *Proceedings of the IEEE*, 83(11):1497–1529, 1995.

[16] Y. C. Chen and A. D. Domínguez-García. Assessing the impact of wind variability on power system small-signal reachability. In *Proc. of the International Conference on System Sciences*, pages 1–8, 2011.

[17] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, 2003.

[18] E. A. Cross and I. M. Mitchell. Level set methods for computing reachable sets of systems with differential algebraic equation dynamics. In *Proc of the American Control Conference*, pages 2260–2265, 2008.

[19] T. Dang, A. Donze, and O. Maler. Verification of analog and mixed-signal circuits using hybrid systems techniques. In *Formal Methods for Computer Aided Design*, LNCS 3312, pages 21–36. Springer, 2004.

[20] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *Hybrid Systems: Computation and Control*, pages 11–19, 2010.

[21] H. Elmqvist and S. E. Mattsson. Modelica – the next generation modeling language – an international design effort. In *Proc. of the World Congress of System Simulation*, 1997.

[22] G. Fábián, D. A. van Beek, and J. E. Rooda. Index reduction and discontinuity handling using substitute equations. *Mathematical and Computer Modelling of Dynamical Systems*, 7(2):173–187, 2001.

[23] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Proc. of the 23rd International Conference on Computer Aided Verification*, LNCS 6806, pages 379–395. Springer, 2011.

[24] C. W. Gear. Differential-algebraic equation index transformations. *SIAM Journal on Scientific Computing*, 9(1):39–47, 1988.

[25] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, LNCS 3414, pages 291–305. Springer, 2005.

[26] A. Girard and C. Le Guernic. Efficient reachability analysis for linear systems using support functions. In *Proc. of the 17th IFAC World Congress*, pages 8966–8971, 2008.

[27] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 257–271. Springer, 2006.

[28] Z. Han and B. H. Krogh. Reachability analysis of nonlinear systems using trajectory piecewise linearized models. In *Proc. of the American Control Conference*, pages 1505–1510, 2006.

[29] J. Hoefkens, M. Berz, and K. Makino. *Automatic Differentiation: From Simulation to Optimization*, chapter Efficient High-Order Methods for ODEs and DAEs, pages 343–348. Springer, 2001.

[30] J. Hoefkens, M. Berz, and K. Makino. *Scientific Computing, Validated Numerics, Interval Methods*, chapter Verified High-Order Integration of DAEs and Higher-Order ODEs, pages 281–292. Springer, 2001.

[31] L. Jaulin, M. Kieffer, and O. Didrit. *Applied Interval Analysis*. Springer, 2006.

[32] L. Jin, H. Liu, R. Kumar, J. D. McCalley, N. Elia, and V. Ajjarapu. Power system transient stability design using reachability based stability-region computation. In *Proc. of the 37th Annual North American Power Symposium*, pages 338–343, 2005.

[33] W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61:47–67, 1998.

[34] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.

[35] G. Lafferriere, G. J. Pappas, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Symbolic Computation*, 32:231–253, 2001.

[36] A. S. Miner and G. Ciardo. Efficient reachability set generation and storage using decision diagrams. In *Proc. of the 20th International Conference on Applications and Theory of Petri Nets*, LNCS 1639, pages 6–25. Springer, 1999.

[37] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50:947–957, 2005.

[38] I. M. Mitchell and Y. Susuki. Level set methods for computing reachable sets of hybrid systems with differential algebraic equation dynamics. In *Hybrid Systems: Computation and Control*, LNCS 4981, pages 630–633. Springer, 2008.

[39] M. Moechtar, T. C. Cheng, and L. Hu. Transient stability of power system – a survey. In *Proc. of the WESCON conference*, pages 166–171, 1995.

[40] N. S. Nedialkov and M. von Mohrenschildt. Rigorous simulation of hybrid dynamic systems with symbolic and interval methods. In *Proc. of the American Control Conference*, pages 140–147, 2002.

[41] L. R. Petzold. Differential/algebraic equations are not ODEs. *SIAM Journal on Scientific and Statistical Computing*, 3(3):367–384, 1982.

[42] A. Puri, P. Varaiya, and V. Borkar. ϵ-approximation of differential inclusions. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 2892 – 2897, 1995.

[43] C. V. Ramamoorthy. Analysis of graphs by connectivity considerations. *Journal of the ACM*, Volume 13 Issue 2, April 1966:211–222, 1966.

[44] N. Ramdani and N. S. Nedialkov. Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint-propagation techniques. *Nonlinear Analysis: Hybrid Systems*, 5(2):149–162, 2010.

[45] M. Ribbens-Pavella and F. J. Evans. Direct methods for studying dynamics of large-scale electric power systems – a survey. *Automatica*, 21(1):1–21, 1985.

[46] P. Schavemaker and L. van der Sluis. *Electrical Power System Essentials*. Wiley, 2008.

[47] L. F. Shampine, M. W. Reichelt, and J. A. Kierzenka. Solving index-1 DAEs in MATLAB and Simulink. *SIAM Review*, 41:538–552, 1999.

[48] G. V. Smirnov. *Introduction to the Theory of Differential Inclusions*. American Mathematical Society, 2002.

[49] Y. Susuki, T. Sakiyama, T. Ochi, T. Uemura, and T. Hikihara. Verifying fault release control of power system via hybrid system reachability. In *Proc. of the 40th North American Power Symposium*, 2008.

[50] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification and control of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[51] University of Washington. Power systems test case archive. http://www.ee.washington.edu/research/pstca/.

Matthias Althoff is assistant professor in computer science at Technische Universität München, Germany. He received the diploma engineering degree in Mechanical Engineering in 2005, and the Ph.D. degree in Electrical Engineering in 2010, both from Technische Universität München, Germany. From 2010 to 2012 he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013 an assistant professor at Technische Universität Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, automated vehicles, and power systems.

Bruce H. Krogh is professor of electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA and is currently Director of Carnegie Mellon University in Rwanda. He was a past Associate Editor of the IEEE Transactions on Automatic Control and Discrete Event Dynamic Systems: Theory and Applications, and founding Editor-in-Chief of the IEEE Transactions on Control Systems Technology. His current research interests include synthesis and verification of embedded control software, discrete event and hybrid dynamic systems, and distributed control strategies for the smart grid and other energy-related applications.