2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Level Set Methods for Computing Reachable Sets of Systems with Differential Algebraic Equation Dynamics

Elizabeth Ann Cross and Ian M. Mitchell
Department of Computer Science,
University of British Columbia,
2366 Main Mall, Vancouver, BC, V6T 1Z4
{ecross,mitchell}@cs.ubc.ca

***Abstract*— Most existing algorithms for approximating the reachable sets of continuous systems assume an ordinary differential equation model of system evolution. In this paper we adapt such an existing algorithm—one based on level set methods and the Hamilton-Jacobi partial differential equation—in two distinct ways to work with systems modeled by index one differential algebraic equations (DAEs). The first method works by analytic projection of the dynamics onto the DAE's constraint manifold, while the second works in the full dimensional state space. The two schemes are demonstrated on a nonlinear power system voltage safety problem.**

I. Introduction

The reachable set or tube is a very general tool for verifying the safe operation of a system; unfortunately, it can rarely be determined exactly for continuous or hybrid systems. Design of approximation algorithms depends critically on the type of mathematical model chosen to describe the evolution of the system. In this paper, we adapt reachability algorithms based on level set methods and the Hamilton-Jacobi (HJ) partial differential equation (PDE) to the class of continuous state and time models called differential algebraic equations (DAEs). In particular, we focus on index one DAEs, which can be alternatively viewed as ordinary differential equations (ODEs) evolving on constraint manifolds.

The contribution of this paper is two algorithms for approximating the backward reachable set or tube of systems described by nonlinear DAEs of index one. The first algorithm operates on the constraint manifold, and so uses a lower dimensional state space but requires that the manifold be explicitly parameterized. The second and more general algorithm works in the full dimensional state space. While we use a simple toy example to explain each of the two schemes, we conclude the paper with a power system voltage safety analysis to demonstrate their practical application to a complex but low dimensional nonlinear model. The schemes are successful at approximating reach tubes; however, they are both subject to the dimensional scaling problems common to grid-based reachability algorithms. That said, these results are useful not only for computation, but also from a theoretical perspective. The two algorithms demonstrate that HJ PDEs can be used to describe the evolution of reach sets and tubes for DAEs; consequently, any theoretical analysis arising from HJ PDEs for reachability of ODE models can likely be applied to DAE models as well.

II. Background

A. Reach Sets and Tubes

Consider a model of a system operating in a state space $\mathbb{S}$ of dimension $d_{\mathbb{S}}$ (typically $\mathbb{S} = \mathbb{R}^{d_{\mathbb{S}}}$) with a set of known initial states $I \subset \mathbb{S}$ and a set of known unsafe states $T \subset \mathbb{S}$ (the "target" set). Trajectories of the model will be denoted by $x(\cdot) : \mathbb{T} \to \mathbb{S}$, where $\mathbb{T} \subset \mathbb{R}$ is the time interval over which the trajectory exists, and $x(0) = x_0$. The system is considered safe as long as there does not exist a $t \in \mathbb{T}$ and $x_0 \in I$ such that $x(t) \in T$; in other words, no trajectory exists which travels from the initial states to the unsafe states.

One common tool for verifying the safety of a system's model is the reachable set or tube. For the input-free system described above, the backwards reach set and reach tube (we use "reachable" and "reach" interchangably) are defined respectively as

$$B(T,t) \triangleq \{x_0 \in \mathbb{S} \mid \exists \hat{x} \in T, x(t) = \hat{x}\},$$
$$B(T,[0,t]) \triangleq \{x_0 \in \mathbb{S} \mid \exists \hat{x} \in T, \exists s \in [0,t], x(s) = \hat{x}\}.$$

The backwards reach set at time t is the set of states which give rise to trajectories which reach T in exactly t time units. The backward reach tube over time interval $[0,t]$ is the set of states which give rise to trajectories which reach T at any time $s \in [0,t]$. Once the backward reach tube is determined, then the system is safe until at least time t if $B(T,[0,t]) \cap I = \emptyset$. In the remainder of this paper we will use reach tubes exclusively, although the algorithms can be equally well applied to approximate the reach set. Further discussion of the connections between backward reach sets and tubes, as well as the related forward reach sets and tubes, can be found in [1] and the citations therein.

B. Approximating Reach Tubes with Level Set Methods

Except for simple or contrived examples, the backward reach tube cannot be determined analytically. In this paper we will use the method described in [2] to approximate the backward reach tube, although we restrict our presentation to systems without inputs so that we can use a simpler notation for trajectories. Under this restriction, the method works on a system modeled by the ODE $dx(t)\,/\,dt = \dot{x}(t) = f_{\mathbb{S}}(x(t))$ subject to initial conditions $x(0) = x_0$. The dynamics $f_{\mathbb{S}} : \mathbb{S} \to \mathsf{T}\mathbb{S}$ are assumed to be Lipschitz continuous and bounded.

To determine the backward reach tube, we must be given an implicit surface function $\phi_0 : \mathbb{S} \to \mathbb{R}$ such that $T = \{x \in \mathbb{S} \mid \phi_0(x) \leq 0\}$. The backward reach tube is then implicitly defined by the function $\phi : \mathbb{S} \times \mathbb{T} \to \mathbb{R}$, where

$$B(T, [0, t]) = \{x \in \mathbb{S} \mid \phi(x, -t) \leq 0\}. \tag{1}$$

The implicit surface function ϕ is the viscosity solution to the HJ PDE

$$D_s\phi(x, s) + \min\left[0, D_x\,\phi(x, s) \cdot f_{\mathbb{S}}(x)\right] = 0 \tag{2}$$

solved backwards in time from $s = 0$ with terminal value $\phi(x, 0) = \phi_0(x)$.

Of course, analytic solutions for HJ PDEs such as (2) are not generally available either. Instead, we use the group of numerical algorithms known collectively as level set methods to approximate the solution. In particular, we use the publicly available implementation TOOLBOXLS [3], [4].

C. Differential Algebraic Equations

ODEs are the most common model for the evolution of a continuous system's state, but far from the only one. For many systems, the most convenient model involves a mixture of differential and algebraic equations, giving rise to a DAE. For the DAE models of interest in this paper we can partition the state space $\mathbb{S}$ into those states $\mathbb{D}$ directly governed by differential equations and those states $\mathbb{A}$ directly governed by algebraic constraints. Let $d_{\mathbb{D}}$ be the dimension of $\mathbb{D}$ and $d_{\mathbb{A}}$ be the dimension of $\mathbb{A}$. For a state $x \in \mathbb{S} = \mathbb{D} \times \mathbb{A}$ we will write $x = (y, z)$ where $y \in \mathbb{D}$ and $z \in \mathbb{A}$. Then the system model takes the form of a semi-explicit DAE

$$\dot{y}(s) = f_{\mathbb{D}}(y(s), z(s)) \tag{3}$$

$$0 = g(y(s), z(s)) \tag{4}$$

where $f_{\mathbb{D}} : \mathbb{D} \times \mathbb{A} \to \mathsf{T}\mathbb{D}$, $g : \mathbb{D} \times \mathbb{A} \to \mathbb{R}^{d_{\mathbb{A}}}$ and the initial conditions are $y(0) = y_0$ and $z(0) = z_0$.

The index of a DAE is a metric which in some sense describes how far the DAE is from being an ODE; ODEs are defined to be DAEs of index zero. Mathematically, the index is the number of times which the algebraic equations must be differentiated in order to retrieve a purely differential model of the system. In this paper we will restrict ourselves to DAEs of index one. To determine an almost equivalent ODE for (4), we differentiate with respect to t, apply the chain rule, and rearrange to find

$$\dot{z} = (D_z\, g(y, z))^{-1}(D_y\, g(y, z) f_{\mathbb{D}}(y, z)) \triangleq f_{\mathbb{A}}(y, z) \tag{5}$$

subject to the same initial conditions. It should be noted that while solutions of (3) and (4) are solutions of (3) and (5), the latter also admits any trajectories which satisfy (3) and hold $g(y, z) = c$ for any constant c. In practical terms, numerical approximation of the solution of (3) and (4) by applying an ODE integrator to (3) and (5) is a poor idea because the small errors introduced by each timestep of the integrator cause gradual violation of the algebraic constraint $g(y, z) = 0$. Fortunately, a number of numerical schemes are available for accurately approximating the solution of DAEs with index one [5]; for example, the sample trajectories used in this paper were generated with MATLAB's `ode15s` [6].

As a simple example of a DAE system, in the next two sections we use the two dimensional model

$$\begin{aligned} \dot{y} &= -(3/2 + z) = f_{\mathbb{D}}(y, z), \\ 0 &= z - \sin(\pi y) = g(y, z), \end{aligned} \tag{6}$$

where $y \in \mathbb{D} = \mathbb{R}$ and $z \in \mathbb{A} = \mathbb{R}$. For visualization purposes, we will let z be the horizontal coordinate and y be the vertical coordinate. The almost equivalent ODE system is given by

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -(3/2 + z) \\ -\pi\cos(\pi y)(3/2 + z) \end{bmatrix} = \begin{bmatrix} f_{\mathbb{D}}(y, z) \\ f_{\mathbb{A}}(y, z) \end{bmatrix}.$$

Qualitatively, the constraint in (6) is a sine curve $z = \sin(\pi y)$, along which the trajectory moves faster when z is more positive. For this system we will determine the backward reach tube starting from the target set $T = \{(y, z) \mid y \leq 0 \wedge g(y, z) = 0\}$.

D. Related Work

Many other schemes have been used for approximating reach sets and tubes; for reasons of space we include here only the work and issues most relevant to systems modeled by DAEs. Extensive discussions of and citations to the full variety of reachability schemes for continuous and hybrid systems can be found in [1], [2].

Almost all of the previous work on continuous reachability has focused on systems whose state evolution is described by ordinary differential or difference equations. One exception is [7], which adapts the d/dt tool [8] to a DAE setting by replacing the standard ODE integration schemes in d/dt with a standard DAE integration scheme: the equivalent ODE (3) and (5) is solved at each timestep, numerical errors in this process cause drift away from the constraint manifold, and so the result is projected back onto the manifold to ensure that (4) holds [5]. This approach could also be applied to any other reachability algorithm based on explicitly following system trajectories ("Lagrangian" algorithms), but the projection step inevitably introduces error. An advantage of the techniques examined here is that no such projection is needed—the reach tube always satisfies (4).

Although we are not aware of any viability work explicitly addressing DAE-like systems, it should be straightforward to adapt viability kernel algorithms [9] to handle systems such as those examined here. The primary challenge of a direct adaptation in the full dimensional state space will be accurate approximation of the lower dimensional constraint manifold.

III. COMPUTATIONS ON THE ALGEBRAIC MANIFOLD

The HJ PDE based approach to reach tubes explained in section II-B makes few restrictions on the dynamics $f_{\mathbb{S}}$. Consequently, one approach to determining the reach tube is to parameterize the constraint manifold (4), apply an appropriate change of variables to the differential dynamics (3), and solve the HJ PDE on the manifold.

Denote the coordinate system on the constraint manifold by $x \in \mathbb{M} \subset \mathbb{R}^{d_{\mathbb{D}}}$. Let $y = u(x)$ be a diffeomorphism

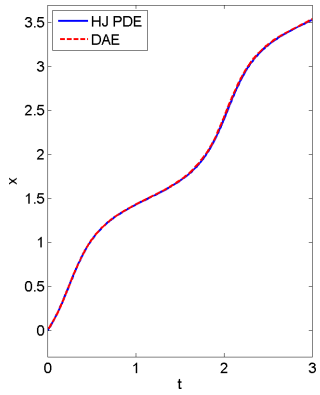


Fig. 1. Zero level set of the implicit surface function $\phi(x,t)$ solving (8), which approximates the backward reach tube on the constraint manifold of the toy system (6). Also plotted is a numerical approximation of the solution of the DAE mapped into the manifold coordinate system. Qualitatively, the two solutions are indistinguishable.

mapping from the constraint manifold coordinates $\mathbb{M}$ to the differential subspace $\mathbb{D}$. Also, let $z = v(x)$ be a continuous function from $\mathbb{M}$ to $\mathbb{A}$. Using the chain rule, $\dot{y} = D_x\, u(x)\dot{x}$, so the dynamics on the constraint manifold are given by

$$\begin{aligned}\dot{x} &= (D_x\, u(x))^{-1}\dot{y},\\ &= (D_x\, u(x))^{-1} f_{\mathbb{D}}(u(x), v(x)) \triangleq f_{\mathbb{M}}(x),\end{aligned} \tag{7}$$

where $D_x\, u(x)$ is nonsingular because u is a diffeomorphism. Consequently, we can easily solve the HJ PDE (2) in the parameterized coordinate system with the substitution $f_{\mathbb{S}}(x) \leftarrow f_{\mathbb{M}}(x)$, and then the reach tube is given by (1). Given $u(x)$ and $v(x)$, the greatest challenge to solving (2) is likely to be determination of initial conditions $\phi_0(x)$ which represent T on the constraint manifold, since T will normally be given in the original (y,z) coordinate system.

To demonstrate this technique, we apply it to the toy system (6). Choosing our manifold coordinate system as $x = y$ we get $y = u(x) = x$, $z = v(x) = \sin(\pi x)$, $D_x\, u(x) = 1$ and $f_{\mathbb{M}}(x) = -(1)^{-1}(3/2 + \sin(\pi x))$. The resulting HJ PDE is

$$D_t\phi(x,t) + \min\left[0, D_x\,\phi(x,t)\cdot -(3/2 + \sin(\pi x))\right] = 0 \tag{8}$$

The initial conditions are $\phi_0(x) = x$. Figure 1 shows the zero isosurface of $\phi(x,t)$ as approximated by applying level set methods to (8). For comparison purposes, the numerical solution of (6) is also plotted in the (x,t) coordinate system.

IV. Computations in the Full State Space

For those DAE systems not amenable to the treatment in the previous section, the alternative is to work in the full state space $\mathbb{S} = \mathbb{D} \times \mathbb{A}$. Now the constraint manifold is a $d_{\mathbb{D}}$ dimensional subspace of the domain, represented by the zero level set (the zero isosurface) of g. Our approach to determining the backward reach tube will be to define an evolution of an implicit surface function ϕ throughout the domain such that the reach tube is the intersection of the zero sublevel set of ϕ and the zero level set of g

$$B(T,[0,t]) = \left\{ (y,z) \in \mathbb{D}\times\mathbb{A} \;\middle|\; \begin{array}{r}\phi(y,z,-t) \le 0\\ \wedge\, g(y,z) = 0\end{array} \right\}. \tag{9}$$

The simplest way of accomplishing this goal is to solve the HJ PDE (2) on $\mathbb{S}$. Since the DAE does not provide a full dimensional set of dynamics, we break the gradient of ϕ into components corresponding to the differential and algebraic subspaces, and plug $f_{\mathbb{D}}$ from (3) and $f_{\mathbb{A}}$ from (5) into (2) to get

$$D_t\phi(y,z,t) + \min\left[0, \begin{array}{l} D_y\,\phi(y,z,t)\cdot f_{\mathbb{D}}(y,z)\\ \quad + D_z\,\phi(y,z,t)\cdot f_{\mathbb{A}}(y,z)\end{array}\right] = 0 \tag{10}$$

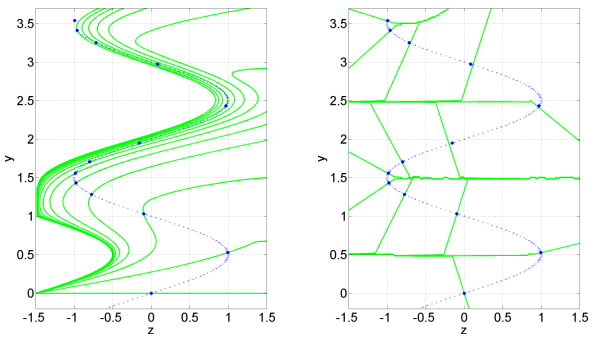
with $\phi(y,z,0) = \phi_0(y,z)$. Numerically, use of $f_{\mathbb{A}}$ does permit some drift in the dynamics away from the constraint manifold; however, since the reach tube approximation is always determined in (9) by comparison with the true constraint manifold, this drift is not a concern.

Unfortunately, this straightforward approach yields disappointing numerical results. In order for the intersection of the zero level sets of ϕ and g to be well-behaved, these level sets should not be too close to parallel. However, even if the level sets of ϕ_0 are nearly perpendicular to those of g, the action of the dynamics $f_{\mathbb{D}}$ and $f_{\mathbb{A}}$ on ϕ can twist these level sets over time and cause them to become nearly parallel.

To avoid this problem, we modify the motion of ϕ with the closest point technique [10]. Because the reach tube is only defined on the constraint manifold, the evolution of ϕ only needs to agree with the underlying DAE on this manifold—away from the manifold, any convenient evolution may be used, as long as it matches the DAE at the manifold. The closest point technique modifies the evolution of ϕ to keep its level sets roughly perpendicular to the constraint manifold in the neighbourhood of the manifold. In addition to keeping the intersection in (9) well-behaved, the gradient of ϕ will be roughly parallel to the constraint manifold because the value of ϕ does not change in directions normal to this manifold; consequently, the drift in dynamics away from the manifold that is mentioned above will be minimized.

To accomplish this goal of maintaining the level sets of ϕ perpendicular to the manifold, define a *closest point* function $\gamma : \mathbb{D}\times\mathbb{A} \to \mathbb{D}\times\mathbb{A}$ which for every point in the domain identifies the closest point on the constraint manifold. Standard level set methods are then used to solve (10) as usual, except that after every timestep we replace the calculated $\phi(y,z,t)$ at each node in the computational grid by $\phi(\gamma(y,z),t)$. Since $\gamma(y,z)$ will not generally be a node in the grid, the value $\phi(\gamma(y,z),t)$ is estimated based on the values of ϕ at nodes adjacent to $\gamma(y,z)$ by an interpolation scheme with an order of accuracy at least one greater than the numerical scheme used to approximate the solution of (10). Further discussion of this approach, including the method by which we create the function γ, can be found in [11].

The results of applying this full dimensional approach to the toy system (6) are shown in figure 2, both with and without the closest point adjustment to the motion of



(a) Without closest point. (b) With closest point.

Fig. 2. Motion of the backward reach tube in the full state space for the toy system (6). The dotted line shows the constraint manifold $g(y,z)=0$, the star symbols are the numerical solutions to the DAE at times $t_i = i/4$ for $i = 0,1,\ldots,12$, (starting from t_0 at the bottom), and the solid lines are the zero level sets of $\phi(y,z,-t_i)$. The intersection of the level set of $\phi(y,z,-t_i)$ and the constraint represents the upper boundary of the backward reach set at t_i, and should align with a star symbol. It is much easier to determine this intersection when the closest point scheme is applied.

Description	Symbol	Value
generator voltage behind transient reactance	E'	
field excitation	E_f	
load bus voltage	E	
generator bus voltage	E_G	
open-circuit transient time constant	T'_{d0}	5
transmission reactance (two routes)	X_1	0.1
d-axis synchronous reactance	X_d	1.2
d-axis transient reactance	X'_d	0.2
time constant of first-order model of AVR	T	1.5
nominal field excitation	E_f^0	1.6
gain constant of first-order model of AVR	K	7
set-point value of generator bus voltage	E_r	1
mechanical input power to generator	P_m	1.0
constant reactive power of load	Q_0	$0.5P_m$
current source of load	H	0
impedance load	B	0
critical value of load bus voltage	E_c	0.7

TABLE I

PHYSICAL MEANING OF VARIABLES AND PARAMETERS IN DAE MODEL (11) FOR THE SINGLE MACHINE-LOAD BUS EXAMPLE [12].

the implicit surface function. By the end of the simulation time, the implicit surface function for the case without closest point is so nearly parallel to the constraint that the intersection of the two is very difficult to see; in contrast, the intersections for the case with closest point remain clearly visible throughout the simulation.

Alternative schemes for evolving implicit surface functions on manifolds have been described; see [10] for a full discussion. For our static manifold problem, we tried several but the closest point scheme proved to be the most effective and efficient, as well as being extensible to manifolds of higher codimension. More details can be found in [11].

V. SINGLE MACHINE-LOAD BUS EXAMPLE

To demonstrate the techniques described above in a more realistic setting, we consider a single machine-load bus system, and in particular a purely continuous single mode version of the hybrid system reach tube problem posed in [13], based on the model given in [12]. The state space is

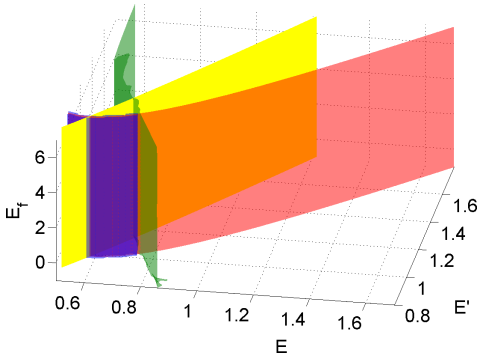


Fig. 3. Important sets and surfaces for the single machine-load bus example. Starting from the back left and moving right along the back edge of the plot, the green (dark grey) surface heading almost directly into the page is the zero level set of ϕ_0 (regions to the left of this surface are $\phi_0 < 0$), the yellow (lightest grey) is the singular surface S, and the red (medium grey) is the constraint surface. The blue (black) region in the front left is the target set T: The intersection of the zero sublevel set of ϕ_0 and the constraint surface. Notice that the singular surface only approaches the constraint surface inside the target set.

$y = (E', E_f) \in \mathbb{D} = \mathbb{R}^2$ and $z = E \in \mathbb{A} = \mathbb{R}$ (note that the prime does not denote a derivative; E' is a separate variable from E). The DAE is

$$\begin{aligned}\begin{bmatrix}\dot{E}'\\ \dot{E}_f\end{bmatrix} &= \begin{bmatrix}\frac{1}{T'_{d0}}\left(\frac{X_d - X'_d}{X'}\frac{E^2 + X'Q(E)}{E'} - \frac{X_1 + X'_d}{X'}E' + E_f\right)\\ \frac{1}{T}\left(-(E_f - E_f^0) - K\left[E_G(E) - E_r\right]\right)\end{bmatrix}\\ &= f_{\mathbb{D}}(E', E_f, E),\\ 0 &= E'^2E^2 - (X'P)^2 - \left[X'Q(E) + E^2\right]^2\\ &= g(E', E),\end{aligned} \tag{11}$$

where

$$\begin{aligned}E_G(E) &= \tfrac{1}{E}\sqrt{(X_1P)^2 + \left[X_1Q(E) + E^2\right]^2},\\ X' &= X_1 + X'_d,\\ P &= P_m,\\ Q(E) &= Q_0 + HE + BE^2.\end{aligned}$$

The physical meaning and values of the variables and parameters are described in table I. The corresponding ODE system replaces the constraint g in (11) with

$$\dot{E} = \frac{E'\dot{E}'E}{2(X'Q(E) + E^2) - E'^2} = f_{\mathbb{A}}(E', E). \tag{12}$$

As discussed in [12], [13], the model (11) has a singular surface

$$S = \{(E', E_f, E) \in \mathbb{D} \times \mathbb{A} \mid D_E\, g(E', E) = 0\}.$$

A view of S and of the constraint surface $g(E', E) = 0$ is shown in figure 3. The model is not an accurate representation of the underlying physics of the system on S, and the DAE (11) is not of index one on this surface. Reviewing the derivation of (12) from (5), we also see that $\dot{E}$ blows up as the state approaches S.

Fortunately, the states in S represent unacceptably low values for the bus voltage E; low voltages can result in

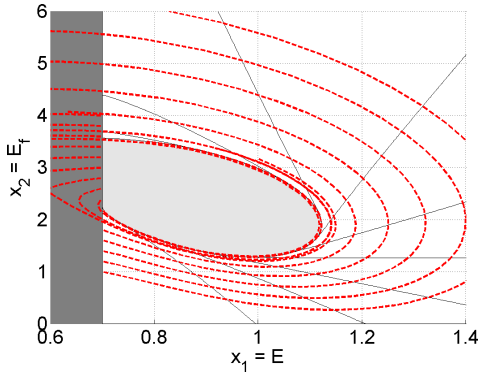


Fig. 4. Approximating the reach tube on the constraint manifold. The general flow on the manifold is counter-clockwise about an equilibrium at $x \approx [\ 0.86\ \ 2.14\ ]^T$. The dark grey region is the initial target set, the thin solid lines show the reach tube's growth every $1/2$ time unit, and the light grey region is the complement of the reach tube after it achieves a fixpoint for $t \gtrsim 5$. For comparison purposes, a number of sample trajectories starting on the boundary of the target set are also shown as dashed lines.

damage or failure of the attached load. Consequently, the unsafe set T for this system is defined as

$$T = \{(E', E_f, E) \in \mathbb{D} \times \mathbb{A} \mid E \leq E_c\},$$

and is also shown in figure 3. We approximate the reach tube for this target using both of the approaches discussed above.

To use the approach from section III, we take manifold coordinate system $x_1 = E$, $x_2 = E_f$ and $\mathbb{M} = \mathbb{R}^2$. For the parameters $H = 0$ and $B = 0$ from table I, we get the simplification $Q(E) = Q_0$ and so

$$u(x) = \begin{bmatrix} \frac{1}{x_1}\sqrt{(X'P)^2 + (X'Q_0 + x_1^2)^2} \\ x_2 \end{bmatrix},$$
$$v(x) = x_1.$$

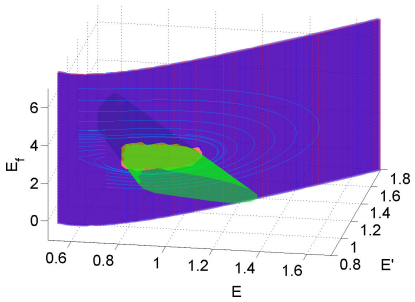
A more complicated but still algebraic formula for $u(x)$ can be derived for the more general $Q(E)$, but we continue with the simpler case here. The resulting dynamics on $\mathbb{M}$ are given by (7), where

$$D_x\, u(x) = \begin{bmatrix} \sigma(x) & 0 \\ 0 & 1 \end{bmatrix},$$
$$\sigma(x) = \frac{x_1^4 - (X'P)^2 - (X'Q_0)^2}{x_1^2\sqrt{(X'P)^2 + (X'Q_0 + x_1^2)^2}}.$$

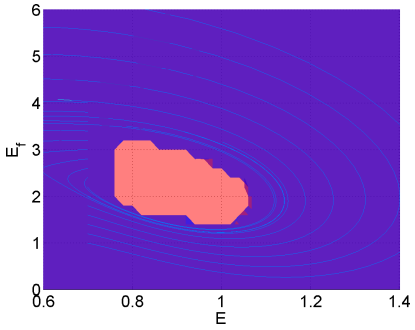
It can be shown that $\sigma(x) \neq 0$ for $x \notin S$, so $D_x\, u(x)$ is nonsingular and $f_{\mathbb{M}}(x)$ is well defined for $x \notin T$. Because the dynamics inside the target set are irrelevant to computation of the reach tube, we apply a smooth approximation of the Heaviside function

$$H(\lambda) = \begin{cases} 0, & \lambda \in (-\infty, -2\epsilon); \\ \frac{1}{2} + \frac{\lambda+\epsilon}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi(\lambda+\epsilon)}{\epsilon}\right), & \lambda \in [-2\epsilon, 0]; \\ 1, & \lambda \in (0, +\infty). \end{cases}$$

to damp the dynamics to zero for $x \in T$ (where $\phi_0(x) \leq 0$), and hence there is no problem caused by the fact that



(a) Same view as figure 3.



(b) Same view as figure 4.

Fig. 5. Approximating the reach tube in the full state space at $t = 5$ (two views). The light blue (light grey) lines in both views are sample trajectories starting on the boundary of the target set. The faceting of the reach tube's boundary arises because of a low resolution grid. The blue (dark grey) surface is the reach tube. Where visible, the green (lighter grey) surface perpendicular to the reach tube is the zero level set of ϕ and the red (light grey) region is the complement of the reach tube (the safe set).

$D_x\, u(x)$ becomes singular for some states inside the target set.

$$f_{\mathbb{M}}(x) = H(\phi_0(x))\,(D_x\, u(x))^{-1}\, f_{\mathbb{D}}(u(x), v(x)). \qquad (13)$$

For the results shown below we used $\epsilon = 2\,\Delta x$, where Δx is the grid node spacing. It is trivial to create an implicit surface function for the target set: $\phi_0(x) = x_1 - E_c$. The results of computing the reach tube for $f_{\mathbb{M}}$ from (13) are shown in figure 4. Only the small light grey region in the center is outside of the reach tube and is thus safe.

To use the approach from section IV, we would like to solve (10) using $f_{\mathbb{D}}$ from (11) and $f_{\mathbb{A}}$ from (12), while applying the closest point function γ to reinitialize ϕ after every timestep. Unfortunately, $f_{\mathbb{A}}$ blows up for states near S, and we need a well defined flow field throughout the full dimensional computational domain to solve (10). We use the same Heaviside function damping trick as above to remove the problem inside the target set. However, in this case there are still portions of S outside T but inside the computational domain. Fortunately, in solving (10) we only need to respect the problem's true dynamics on the constraint manifold, and hence we can choose any flow field to evolve ϕ as long as it agrees with the true flow field on the constraint manifold. Computation of such modified flow fields is known in the level set literature as velocity extension [14]. In this case, we use the closest point function to extend the flow field [10]

and replace (10) with

$$D_t\phi(y,z,t) + \min\begin{bmatrix}0, & D_y\,\phi(y,z,t)\cdot\overline{f_{\mathbb{D}}}(y,z)\\ & +D_z\,\phi(y,z,t)\cdot\overline{f_{\mathbb{A}}}(y,z)\end{bmatrix} = 0.$$

where

$$\overline{f_{\mathbb{D}}}(y,z) = H(\phi_0(\gamma(y,z)))f_{\mathbb{D}}(\gamma(y,z)),$$
$$\overline{f_{\mathbb{A}}}(y,z) = H(\phi_0(\gamma(y,z)))f_{\mathbb{A}}(\gamma(y,z)).$$

The closest point reinitialization $\phi(y,z,t) \leftarrow \phi(\gamma(y,z),t)$ is still applied after each timestep. Results of computing the reach tube in this manner are shown in figure 5.

VI. Conclusion

We have presented two techniques for approximating the reach tube of a system modeled by an index one DAE. While conceptually simple and of lower dimension, the approach in section III is not always applicable. The existence of suitable u and v implies that the original DAE system could be solved as a pure (albeit potentially complicated) ODE of the form (7). In the context of approximating individual trajectories of DAEs, this technique is variously called the indirect or ODE approach and is used in practice [6], but if such a situation holds then why use the DAE model to begin with? In contrast, the approach in section IV may be applied to any DAE of index one; however, because it works in the full dimensional state space, the computational effort will be much greater. This computational tradeoff can clearly be seen in the quality of the reach tube approximation in figures 4 and 5: because it requires one fewer dimension, the former can be computed at a finer resolution in less time than the latter.

The two schemes also differ in their adaptability to other algorithms. The approach examined in section III, which is essentially an analytic projection onto the constraint manifold, could be used with almost any reachability algorithm provided that the algorithm could handle the projected dynamics (which are likely to be nonlinear). The approach examined in section IV makes use of a closest point extension away from the manifold, which is quite specific to implicit surface based representations of the reach tube and hence is unlikely to be useful in other contexts. However, the general idea of evolving a reach tube in the full dimensional state space—making use of conveniently modified dynamics away from the constraint manifold as necessary—and considering only the states where this tube intersects the manifold as truely part of the reach tube might prove productive in other algorithms.

With the addition of a method for handling the state reset that occurs during mode switches, these schemes can be used for reachability in hybrid systems. The method is briefly explained in [15], along with an extension of the single machine-load bus example to a hybrid setting.

Finally, we mention that empirical extension of either technique to systems with inputs is straightforward following the procedures in [2], and as shown there such inputs can be used to robustly treat model uncertainty. However, we do not know whether such an extension might run into theoretical issues unique to DAEs regarding existence and uniqueness of trajectories and/or reach tubes.

Reproducible Research: The code recreating the results using the approach from section III, including code for figures 1 and 4, can be found at [4]; these codes require the ToolboxLS package also located at the same website. Unfortunately, the approach in section IV requires some features not currently available in ToolboxLS, and we apologize that the code for figures 2 and 5 has consequently not been released.

Acknowledgements: The authors would like to thank Prof. Yoshihiko Susuki for introducing us to the single machine-load bus example and its voltage safety problem, as well as discussions on how to simulate and verify the model. This research was supported by the National Science and Engineering Research Council of Canada.

References

[1] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer Verlag, 2007, no. 4416, pp. 428–443.

[2] 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*, vol. 50, no. 7, pp. 947–957, 2005.

[3] I. M. Mitchell and J. A. Templeton, "A toolbox of Hamilton-Jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds. Springer Verlag, 2005, no. 3414, pp. 480–494.

[4] [Online]. Available: http://www.cs.ubc.ca/~mitchell/ToolboxLS

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

[6] L. F. Shampine, M. W. Reichelt, and J. A. Kierzenka, "Solving index-1 DAEs in MATLAB and Simulink," *SIAM Review*, vol. 41, no. 3, pp. 538–552, 1999.

[7] T. Dang, A. Donzé, and O. Maler, "Verification of analog and mixed-signal circuits using hybrid system techniques," in *Formal Methods in Computer-Aided Design*, ser. Lecture Notes in Computer Science, A. J. Hu and A. K. Martin, Eds. Springer Verlag, 2004, no. 3312, pp. 21–36.

[8] E. Asarin, T. Dang, and O. Maler, "d/dt: A verification tool for hybrid systems," in *Proceedings of the IEEE Conference on Decision and Control*, Orlando, FL, 2001, pp. 2893–2898.

[9] P. Saint-Pierre, "Approximation of the viability kernel," *Applied Mathematics and Optimization*, vol. 29, pp. 187–209, 1994.

[10] S. J. Ruuth and B. Merriman, "A simple embedding method for solving partial differential equations on surfaces," *Journal of Computational Physics*, vol. 227, pp. 1943–1961, 2008.

[11] E. A. Cross, "Solving reachable sets on a manifold," Master's thesis, Department of Computer Science, University of British Columbia, August 2007.

[12] V. Venkatasubramanian, H. Schättler, and J. Zaborszky, "Voltage dynamics: Study of a generator with voltage control, transmission, and matched MW load," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1717–1733, November 1992.

[13] Y. Susuki and T. Hikihara, "Predicting voltage instability of power system via hybrid system reachability analysis," in *Proceedings of the American Control Conference*, New York, NY, 2007, pp. 4166–4171.

[14] D. Adalsteinsson and J. A. Sethian, "The fast construction of extension velocities in level set methods," *Journal of Computational Physics*, vol. 148, pp. 2–22, 1999.

[15] I. M. Mitchell and Y. Susuki, "Level set methods for computing reachable sets of hybrid systems with differential algebraic equation dynamics," to appear in Hybrid Systems Computation and Control 2008, 4 pages.