2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Numerical Computation of Critical System Recovery Parameter Values by Trajectory Sensitivity Maximization

Michael W. Fisher and Ian A. Hiskens

Abstract— **Consider a particular finite-time disturbance applied to a system governed by ordinary differential equations and which possesses a stable equilibrium point. The recovery of the system from a disturbance is a function of the system parameter values. It is an important though challenging problem to identify the system parameter values, called critical parameter values, for which the system is just marginally unable to recover from a particular disturbance. Such critical parameter values correspond to cases where the system state, at the instant when the disturbance clears, is on the boundary of the region of attraction of the stable equilibrium point. The paper proposes novel algorithms for numerically computing critical parameter values, both for one and arbitrary dimensional parameter spaces. In the latter case, the algorithm computes the critical parameter values that are nearest to a given point in parameter space. The key idea underpinning the algorithms is that on the boundary of the region of attraction, the trajectory becomes infinitely sensitive to small changes in parameter value. Therefore, critical parameter values are found by varying parameters so as to maximize trajectory sensitivities. The algorithms are demonstrated using a fourth-order power system test case.**

I. Introduction

Engineered systems experience disturbances which have the potential to disrupt desired operation. The ability of the system to recover from a particular finite-time disturbance, such as a fault on a certain transmission line in a power system, to a desired operating point depends on the system parameters. From a systems perspective, the disturbance can be thought of as a parameter dependent initial condition to the post-disturbance dynamical system (which itself is parameter dependent). It is an important and challenging problem to determine the parameter values for which the system is just marginally unable to recover from a particular disturbance, which we term *critical parameter values*. Solving this problem is of value for many applications, such as assessing fault vulnerability in power systems. This paper develops novel, theoretically motivated algorithms for efficient numerical computation of critical parameter values in both one and arbitrary dimensional parameter space.

The setting is a parameter dependent system of ordinary differential equations (ODEs) which possesses a parameter dependent stable hyperbolic equilibrium point that represents desired operation. Let p denote a set of parameters, $z(p)$ the post-disturbance initial condition corresponding to parameter value p, $R(p)$ the (post-disturbance) region of attraction of the stable equilibrium point corresponding to parameter value p, and $B(p)$ the topological boundary of $R(p)$. For certain parameter values p, $z(p)$ lies inside $R(p)$; these are the parameter values for which the system is able to recover from the disturbance. For other parameter values, $z(p)$ lies outside the closure of $R(p)$, and the system is not able to recover from the disturbance in these cases. The boundary case occurs for parameter values p such that $z(p)$ lies in $B(p)$; these are the parameter values for which the system is marginally unable to recover from the disturbance, i.e. the critical parameter values.

Prior work has sought to develop algorithms for numerically computing critical parameter values by exploiting the topology of $B(p)$. Classical algorithms [1]–[4] have several limitations: they require the existence of an energy function for the system whose sublevel sets approximate $R(p)$ well, the identification of a particular unstable equilibrium point in state space with desired properties, and that the post-disturbance dynamics are independent of the parameter p. However, energy functions have only been developed for simplified power system models, identification of the required equilibrium point is computationally intractable, and almost all parameters of interest influence post-disturbance dynamics. Therefore, the classical algorithms offer limited benefits for practical power system models.

More recent algorithms [5], [6] for critical parameter value computation do not require the existence of an energy function and can handle parameter dependent post-disturbance dynamics. However, they still require identification of a particular unstable equilibrium point, which is computationally challenging in practice. In contrast, the algorithms developed here do not require the existence of an energy function, can handle parameter dependent post-disturbance dynamics, and also do not require identification of a particular unstable equilibrium point. Therefore, they represent practical algorithms for critical parameter value computation.

The algorithms in this paper follow from the fact that on $B(p)$ the trajectory becomes infinitely sensitive to small changes in parameter value. This is because infinitesimal differences in parameter value determine whether the trajectory converges back to the equilibrium point or goes elsewhere. The key idea of the algorithms presented later is to vary parameter values so as to maximize this trajectory sensitivity. In doing so, $z(p)$ will approach $B(p)$, causing the parameter values to approach their critical values. In practice, it turns out to be preferable to minimize the inverse of the trajectory sensitivities rather than to perform the direct maximization, so this variation is implemented. As critical parameters are

M. W. Fisher and I. A. Hiskens are with the EECS Department, University of Michigan, Ann Arbor, MI USA, {fishermw,hiskens}@umich.edu.

This work was partially supported by the National Science Foundation through grant ECCS-1810144.

978-1-7281-1398-2/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

not unique, the algorithms are designed to find the nearest critical parameter value to a given initial parameter set.

The paper is organized as follows. Section II introduces notation and provides theoretical motivation. Section III presents the algorithms, while Section IV discusses the efficient computation of trajectory sensitivities, which are needed for the algorithms. Section V illustrates the algorithms on a test case. Concluding remarks are provided in Section VI.

II. Theory

We consider a system of ordinary differential equations (ODEs):

$$\dot{y} = g(y,p)$$
$$\dot{p} = 0,$$

where g is C^3, $y \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ for $n, m \geq 1$. We use $\dot{y}$ to denote the time derivative of y, and similarly for p and elsewhere in the paper. Let $x = [y^\intercal \; p^\intercal]^\intercal$ and $f(x) = [g(y,p)^\intercal \; 0^\intercal]^\intercal$ where 0 has dimension m by 1, and where for any matrix M, we let $M^\intercal$ denote the transpose of M. Then we can rewrite the dynamics as:

$$\dot{x} = f(x). \tag{1}$$

Let f_i denote the i-th entry of f and x_j denote the j-th entry of x. There exists a flow ϕ such that $\phi(z,t)$ represents the integral of (1) from initial condition z for time t. Let $x(t) = \phi(z,t)$ when the initial condition $x(0) = z$ is understood, and let $\phi_i(z,t)$ denote the i-th component of the flow.

Fix $p_0 \in \mathbb{R}^m$ and suppose (1) possesses a stable hyperbolic equilibrium point $x^0(p_0)$. Then there exists an open neighborhood J of p_0 in $\mathbb{R}^m$ and a C^1 function $x^0 : J \to \mathbb{R}^{m+n}$ such that $x^0(p)$ is a stable hyperbolic equilibrium point near $x^0(p_0)$. For $p \in J$, let $R(p)$ denote the region of attraction of $x^0(p)$ and let $B(p)$ denote the topological boundary of $R(p)$. Shrinking J if necessary, we assume (1) satisfies Assumptions 1 – 4 of [7], which are technical assumptions that are satisfied for a large class of ODEs. In particular, most of those assumptions are true generically for C^1 ODEs (see [7] for a more complete discussion). Since $x^0(p)$ is a stable hyperbolic equilibrium point, it possesses a local stable manifold $W(p)$ such that any trajectory which converges to $x^0(p)$ must enter $W(p)$ in finite time, and for any $z \in W(p)$ and any $t > 0$, $\phi(z,t) \in W(p)$ and $\left|\left|\frac{\partial \phi}{\partial z}(z,t)\right|\right| < 1$ [8].

A finite time disturbance is modeled by $z : J \to \mathbb{R}^{m+n}$, a C^1 map such that $z(p)$ denotes the system state at the instant when the disturbance clears, also known as the post-disturbance initial condition to (1). Let $J_R \subset J$ be $\{p \in J : z(p) \in R(p)\}$ and let $J_B \subset J$ be $\{p \in J : z(p) \in B(p)\}$. Then J_B is the set of critical parameter values. Let $\chi(t,p) := \frac{\partial \phi(z(p),t)}{\partial p}$ and let $\Phi : J \to [0,\infty]$ be $\Phi(p) := \sup_{t\geq 0} ||\chi(t,p)||^2$ where $||\cdot||$ is any matrix norm and $t = 0$ is defined to be the time at which the disturbance clears. Assume the initial conditions cross the region of attraction boundary transversely in an appropriate sense. Proof sketches of the following Lemmas are provided for brevity.

Lemma 1. Φ is well-defined, finite, and continuous over J_R.

Proof Sketch of Lemma 1. For $p \in J_R$ the flow of $z(p)$ converges to $x^0(p)$, so it enters the local stable manifold $W(p)$ in finite time T. For any time $t > T$, splitting the orbit into the interval from 0 to T and then the interval from T to t, using $\chi(t,p) = \frac{\partial \phi}{\partial z}\frac{\partial z}{\partial p}$, and by the chain rule we have

$$\begin{aligned} ||\chi(t,p)|| &= \left|\left|\frac{\partial \phi}{\partial z}(\phi(z(p),T),t-T)\frac{\partial \phi}{\partial z}(z(p),T)\frac{\partial z}{\partial p}(p)\right|\right| \\ &< \left|\left|\frac{\partial \phi}{\partial z}(z(p),T)\frac{\partial z}{\partial p}(p)\right|\right|, \end{aligned}$$

since $\phi(\phi(z(p),T),t-T) \in W(p)$ and so $\left|\left|\frac{\partial \phi}{\partial z}(\phi(z(p),T),t-T)\right|\right| < 1$. Hence, $||\chi(t,p)||$ must attain a maximum at some finite time in $[0,T]$. As the same inequality holds for $\hat{p}$ near p, $||\chi(t,\hat{p})||$ will also attain a maximum in $[0,T]$. By continuity of the maximum over a compact set, $\Phi(\hat{p})$ will therefore be near $\Phi(p)$. □

Lemma 2. For any $p^* \in J_B$, $\Phi(p^*) = \infty$. Furthermore, if $\{p_s\}_{s=1}^{\infty}$ is a sequence in J_R with $p_s \to p^*$ then $\lim_{s\to\infty} \Phi(p_s) = \infty$.

Proof Sketch of Lemma 2. If $p^* \in J_B$, by [7, Theorem 1] the flow of $z(p^*)$ converges to a hyperbolic equilibrium point or periodic orbit in $B(p^*)$. Therefore, the Inclination Lemma [9] can be applied to show that $\lim_{t\to\infty} \|\chi(t,p^*)\| = \infty$. If $\{p_s\}_{s=1}^{\infty}$ is as in the statement of the Lemma, then using that $\lim_{t\to\infty} \|\chi(t,p^*)\| = \infty$ and continuity of $\chi(t,p)$, it follows that $\lim_{s\to\infty} \Phi(p_s) = \infty$. □

Let $G(p) = \inf_{t\geq 0} \frac{1}{||\chi(t,p)||_F^2}$ where $||\cdot||_F$ is the Frobenius matrix norm which can be expressed by $||M||_F^2 := \sum_j M_{:,j} \cdot M_{:,j}$ where $M_{:,j}$ is the j-th column of M and $v \cdot w$ denotes the dot product of vectors v and w. We define G in this manner because empirical observations from numerical experiments have indicated that G is typically convex, so minimizing G is easier than maximizing Φ, even though the results are equivalent in theory.

Theorem 1. G is well-defined, strictly positive, and continuous over J_R, $G(p^*) = 0$ for all $p^* \in J_B$, and if $\{p_s\}_{s=1}^{\infty}$ is a sequence in J_R such that $p_s \to p^* \in J_B$ then $\lim_{s\to\infty} G(p_s) = 0$.

Proof Sketch of Theorem 1. Follows immediately from Lemmas 1 and 2. □

Theorem 1 states that p^* is a critical parameter value if and only if $G(p^*) = 0$. Furthermore, it suggests that critical parameter values can be found by starting from some $p_0 \in J_R$ and varying p so as to send $G(p) \to 0$. This is the theoretical motivation for the algorithms of Section III.

III. Algorithms for Computing Critical Parameter Values

A. One-Dimensional Parameter Space

Let p be a one-dimensional, real-valued parameter. Assume that the initial condition $z = z(p)$ is a function of p. Define

$$H(t,p) = \frac{1}{\chi(t,p) \cdot \chi(t,p)}$$

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

$$G(p) = \inf_{t \geq 0} H(t,p),$$

where $\chi(t,p) \in \mathbb{R}^{(n+m)\times 1}$ because $p \in \mathbb{R}^1$. Then by Theorem 1, p^* is a critical parameter value if and only if it satisfies:

$$G(p^*) = 0. \tag{2}$$

To find the critical parameter p^*, (2) will be solved using Newton-Raphson, which will require the derivative $\frac{dG(p)}{dp}$. To compute this, first note that

$$\frac{\partial H(t,p)}{\partial p} = -\frac{2\frac{\partial \chi(t,p)}{\partial p} \cdot \chi(t,p)}{\left(\chi(t,p)\cdot\chi(t,p)\right)^2}.$$

Let $\hat{t}(p) = \text{argmin}_{t \geq 0} H(t,p)$. Then, by Theorem 1, if p is not a critical parameter value then $\hat{t}(p)$ is finite so $G(p) = H(\hat{t}(p),p)$. Differentiating with respect to p yields

$$\begin{aligned} DG(p) = \frac{dG(p)}{dp} &= \frac{\partial}{\partial p} H(\hat{t}(p),p) \\ &= \frac{\partial H}{\partial t}(\hat{t}(p),p)\frac{d\hat{t}(p)}{dp}(p) + \frac{\partial H}{\partial p}(\hat{t}(p),p) \\ &= \frac{\partial H}{\partial p}(\hat{t}(p),p), \end{aligned} \tag{3}$$

where the final equality follows since $\frac{\partial H}{\partial t}(\hat{t}(p),p) = 0$ because $\hat{t}(p)$ is the time when $H(t,p)$ achieves a minimum in time, and $\frac{\partial H}{\partial t} = 0$ at an extremum point. Therefore, (2) is solved iteratively by the following standard Newton-Raphson update, where p^s denotes the parameter value of the current iteration, and p^{s+1} the value of the next iteration:

$$p^{s+1} = p^s - DG(p^s)^{-1}G(p^s).$$

As $DG(p) = \frac{\partial H}{\partial p}(\hat{t}(p),p)$ and $G(p) = H(\hat{t}(p),p)$, by the formulas for H and $\frac{\partial H}{\partial p}$, computation of G and DG therefore requires knowledge of $\chi(t,p)$ and $\frac{\partial \chi(t,p)}{\partial p}$. These partial derivatives with respect to parameter, known as trajectory sensitivities, can be efficiently computed numerically as a byproduct of the underlying integration scheme, as discussed in detail in Section IV. Then $\hat{t}(p)$ is observed from the output of the integration, and the values of $G(p)$ and $DG(p)$ can be computed. As the algorithm proceeds, $G(p^s)$ approaches zero, causing p^s to approach a critical parameter value.

B. Multi-Dimensional Parameter Space

Let p be a set of parameter values in $\mathbb{R}^m$ for $m > 1$. Let $j,k,l \in \{1,...,m\}$. Define $\chi_{jkl} := \frac{\partial^3 \phi(z(p),t)}{\partial p_j \partial p_k \partial p_l}$, and define χ_{jk} and χ_k analogously for second and first order partial derivatives, respectively. Note that these are each vectors in $\mathbb{R}^n$. For notational convenience, the dependence on t and p is not explicitly shown. Define $f_x := \frac{\partial f}{\partial x}(\phi(z(p),t))$ and $f_{xx} := \frac{\partial^2 f}{\partial x^2}(\phi(z(p),t))$, noting that f_{xx} is a tensor. Let

$$\begin{aligned} H(t,p) &= \Big(\sum_{k=1}^{m} \chi_k \cdot \chi_k\Big)^{-1} \\ G(p) &= \inf_{t\geq 0} H(t,p) = H(\hat{t}(p),p), \end{aligned}$$

where again $\hat{t}(p) = \text{argmin}_{t\geq 0} H(t,p)$. Fix some $p_0 \in \mathbb{R}^m$. As $m > 1$, there are many critical parameter values that could be found by varying p. Therefore, we seek the critical parameter value that is closest to p_0, which represents the smallest change in parameter space that could lead to failure to recover from the disturbance. So, we wish to solve

$$\min_{p \in \mathbb{R}^m} \frac{1}{2}(p-p_0)^\intercal A (p-p_0) \tag{4}$$

$$\text{s.t. } G(p) - \epsilon = 0, \tag{5}$$

where A is a positive semidefinite weighting matrix (often set to the identity matrix) and $\epsilon > 0$ is small and will be required to ensure feasibility. To solve (4)-(5), we form the Lagrangian $\mathcal{L}(p,\lambda) = \frac{1}{2}(p-p_0)^\intercal A(p-p_0) + \lambda(G(p)-\epsilon)$. Let $F(p,\lambda) := D\mathcal{L}(p,\lambda)$. Any stationary point of $\mathcal{L}(p,\lambda)$, such as a local minimum, must satisfy:

$$0 = D\mathcal{L}(p,\lambda) = \begin{bmatrix} A(p-p_0) + \lambda DG(p)^\intercal \\ G(p)-\epsilon \end{bmatrix} =: F(p,\lambda). \tag{6}$$

Equation (6) is solved iteratively by the following standard Newton-Raphson update, where (p^s, λ^s) denote the corresponding values at the current iteration, and (p^{s+1}, λ^{s+1}) the values at the next iteration:

$$\begin{bmatrix} p^{s+1} \\ \lambda^{s+1} \end{bmatrix} = \begin{bmatrix} p^s \\ \lambda^s \end{bmatrix} - DF(p^s,\lambda^s)^{-1} F(p^s,\lambda^s), \tag{7}$$

where $F(p^s,\lambda^s) = D\mathcal{L}(p,\lambda)$ is given in (6) and DF is given by:

$$DF(p,\lambda) = \begin{bmatrix} A + \lambda D(DG(p)^\intercal) & DG(p)^\intercal \\ DG(p) & 0 \end{bmatrix}. \tag{8}$$

Computation of $DG(p)$ and $D(DG(p)^\intercal)$ first requires several additional derivatives, which are obtained via repeated differentiation of $H(t,p)$ with respect to parameter components and time:

$$\frac{\partial H}{\partial p_j}(t,p) = -\frac{2\sum_{k=1}^m \chi_{jk}\cdot\chi_k}{\left(\sum_{k=1}^m \chi_k \cdot \chi_k\right)^2} \tag{9}$$

$$\begin{aligned} \frac{\partial^2 H}{\partial p_i \partial p_j}(t,p) = &-2H^2\Big(\sum_{k=1}^m \chi_{ijk}\cdot\chi_k + \chi_{jk}\cdot\chi_{ik}\Big) \\ &+ 8H^3\Big(\sum_{k=1}^m \chi_{ik}\cdot\chi_k\Big)\Big(\sum_{k=1}^m \chi_{jk}\cdot\chi_k\Big) \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial^2 H}{\partial p_j \partial t}(t,p) = &-2H^2\Big(\sum_{k=1}^m \dot{\chi}_{jk}\cdot\chi_k + \chi_{jk}\cdot\dot{\chi}_k\Big) \\ &+ 8H^3\Big(\sum_{k=1}^m \dot{\chi}_k\cdot\chi_k\Big)\Big(\sum_{k=1}^m \chi_{jk}\cdot\chi_k\Big) \end{aligned} \tag{11}$$

$$\begin{aligned} \frac{\partial^2 H}{\partial t^2}(t,p) = &\, 8H^3\Big(\sum_{k=1}^m \dot{\chi}_k\cdot\chi_k\Big)^2 \\ &- 2H^2\Big(\sum_{k=1}^m f_{xx}\chi_k\chi_k f + f_x\dot{\chi}_k\cdot\chi_k + \dot{\chi}_k\cdot\dot{\chi}_k\Big), \end{aligned} \tag{12}$$

where $H = H(t,p)$, $f_{xx}\chi_k\chi_k f = f_{lmn}\chi_{mk}\chi_{lk}f_n$ in the notation of Section IV-A, and $\dot{\chi}_k$, $\dot{\chi}_{jk}$ are given by (13)-(14) in Section IV-A. Next we compute the derivatives $DG(p)$

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

and $D(DG(p)^{\intercal})$. First, note that $(DG(p))_j = \frac{\partial G(p)}{\partial p_j}$. By an analogous argument as in the derivation of (3) for the one dimensional parameter case in Section III-A, $(DG(p))_j = \frac{\partial H}{\partial p_j}(\hat{t}(p), p)$. Next, differentiating this equation with respect to p_i gives

$$\begin{aligned}(D(DG(p)^{\intercal}))_{ji} &= \frac{\partial^2 G(p)}{\partial p_i \partial p_j} \\ &= \frac{\partial}{\partial p_i}(DG(p)^{\intercal})_j = \frac{\partial}{\partial p_i}\frac{\partial H}{\partial p_j}(\hat{t}(p), p) \\ &= \frac{\partial^2 H}{\partial p_j \partial t}(\hat{t}(p), p)\frac{\partial \hat{t}(p)}{\partial p_i} + \frac{\partial^2 H}{\partial p_i \partial p_j}(\hat{t}(p), p).\end{aligned}$$

Note that computation of $D(DG(p)^{\intercal})$ requires $\frac{\partial \hat{t}(p)}{\partial p_i}$. We derive this as follows. Since $\hat{t}(p) = \operatorname{argmin}_{t \geq 0} H(t, p)$, $\frac{\partial}{\partial t}H(\hat{t}(p), p) = 0$. Hence, as this function of p is identically zero, differentiating with respect to p_i gives:

$$\begin{aligned}0 =& \frac{\partial}{\partial p_i}\frac{\partial}{\partial t}H(\hat{t}(p), p) = \frac{\partial}{\partial t}\frac{\partial}{\partial p_i}H(\hat{t}(p), p) \\ =& \frac{\partial}{\partial t}\left(\frac{\partial H}{\partial t}(\hat{t}(p), p)\frac{\partial \hat{t}(p)}{\partial p_i} + \frac{\partial H}{\partial p_i}(\hat{t}(p), p)\right) \\ =& \frac{\partial^2 H}{\partial t^2}(\hat{t}(p), p)\frac{\partial \hat{t}(p)}{\partial p_i} + \frac{\partial^2 H}{\partial p_i \partial t}(\hat{t}(p), p).\end{aligned}$$

Hence, solving for $\frac{\partial \hat{t}(p)}{\partial p_i}$ yields

$$\frac{\partial \hat{t}(p)}{\partial p_i} = -\frac{\partial^2 H}{\partial p_i \partial t}(\hat{t}(p), p)\left(\frac{\partial^2 H}{\partial t^2}(\hat{t}(p), p)\right)^{-1}.$$

Substituting this back into the expression for $D(DG(p)^{\intercal})_{ji}$, we obtain

$$\begin{aligned}(DG(p))_j &= \frac{\partial H}{\partial p_j}(\hat{t}(p), p) \\ (D(DG(p)^{\intercal}))_{ji} &= \frac{\partial^2 H}{\partial p_i \partial p_j}(\hat{t}(p), p) \\ &- \frac{\partial^2 H}{\partial p_j \partial t}(\hat{t}(p), p)\frac{\partial^2 H}{\partial p_i \partial t}(\hat{t}(p), p)\left(\frac{\partial^2 H}{\partial t^2}(\hat{t}(p), p)\right)^{-1},\end{aligned}$$

which can be computed from (9)-(12). In turn, the expressions for $DG(p)$ and $D(DG(p)^{\intercal})$ given here can be used to compute F and DF in (8) and (6), respectively. Finally, F and DF are used to perform the Newton-Raphson updates of (7), which drive p^s towards one of the (possibly many locally) closest critical parameter values to p_0.

IV. Trajectory Sensitivities

Trajectory sensitivities are partial derivatives of the flow $\phi(z, t)$ with respect to components of the initial condition z. Note that trajectory sensitivities are therefore functions of time t and initial condition z. The order of a trajectory sensitivity is the number of partial derivatives of the flow that it involves. For a subset of components of the initial condition z that are parameters, trajectory sensitivity computations for (1) yield the partial derivatives of the flow with respect to those parameters. The algorithms described in Section III require the first-, second-, and third-order partial derivatives of the flow with respect to parameters. Therefore, first-, second-, and third-order trajectory sensitivities must be used to compute the necessary partial derivatives with respect to parameters. Computation of these sensitivities is described below. Note that despite vector field discontinuities introduced by the disturbance, continuity of the states implies continuity of the trajectory sensitivities.

A. Time Derivatives

Trajectory sensitivities are computed in practice through numerical integration, as discussed in Section IV-B. To perform the integration, the variational equations describing their evolution are required. This section is devoted to their derivation. The first- and second-order sensitivity variational equations have been derived previously [10], [11]. Derivation of the third-order sensitivity variational equations follows.

Let $i, j, k, l \in \{1, ..., m+n\}$. We define the following notation:

$$\begin{aligned}f_{ij}(x) &:= \frac{\partial f_i(x)}{\partial x_j}, & \chi_{ij}(t) &:= \frac{\partial \phi_i(z,t)}{\partial z_j} \\ f_{ijk}(x) &:= \frac{\partial^2 f_i(x)}{\partial x_j \partial x_k}, & \chi_{ijk}(t) &:= \frac{\partial^2 \phi_i(z,t)}{\partial z_j \partial z_k} \\ f_{ijkl}(x) &:= \frac{\partial^3 f_i(x)}{\partial x_j \partial x_k \partial x_l}, & \chi_{ijkl}(t) &:= \frac{\partial^3 \phi_i(z,t)}{\partial z_j \partial z_k \partial z_l}.\end{aligned}$$

Note that $\chi_{ij}(t)$, $\chi_{ijk}(t)$, and $\chi_{ijkl}(t)$ for all i, j, k, l represent the first-, second-, and third-order trajectory sensitivities, respectively, as functions of time.

Next we will present the time derivatives of the trajectory sensitivities. For clarity and brevity, we use Einstein summation notation, which holds that if an index appears on the right hand side of an equation but not the left, then the index should be summed over on the right hand side. For example, the equation $a_i = b_{ij} + c_{ijk}$ in Einstein summation notation means $a_i = \sum_j b_{ij} + \sum_{j,k} c_{ijk}$. The time derivatives are obtained by repeated differentiation with respect to components of the initial condition, and application of the chain rule, starting with (1):

$$\dot{\chi}_{ij}(t) = f_{im}(x(t))\chi_{mj}(t) \tag{13}$$

$$\dot{\chi}_{ijk}(t) = f_{ilm}(x(t))\chi_{mk}(t)\chi_{lj}(t) + f_{in}(x(t))\chi_{njk}(t) \tag{14}$$

$$\begin{aligned}\dot{\chi}_{ijkl}(t) &= f_{inmo}(x(t))\chi_{ol}(t)\chi_{mk}(t)\chi_{nj}(t) \\ &+ f_{inm}(x(t))\chi_{mkl}(t)\chi_{nj}(t) + f_{inm}(x(t))\chi_{mk}(t)\chi_{njl}(t) \\ &+ f_{ino}(x(t))\chi_{ol}(t)\chi_{njk}(t) + f_{in}(x(t))\chi_{njkl}(t).\end{aligned} \tag{15}$$

Note that once $x(t)$ has been obtained by numerical integration, the time derivatives of the trajectory sensitivities are multilinear functions of the sensitivities. We will see below that this significantly simplifies their numerical integration.

B. Numerical Integration

Earlier work has shown that, using a trapezoidal integration scheme to numerically integrate the underlying dynamics of (1), the first [10] and second [11] order trajectory sensitivities can be efficiently computed as a byproduct of this underlying integration. Here, this efficient computation is extended to third order trajectory sensitivities. Let s denote

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

the current time step of numerical integration. Given the trajectory sensitivities at s, we will compute their values at $s+1$. Let χ^s_{ij}, χ^s_{ijk}, and χ^s_{ijkl} denote the values of $\chi_{ij}(t)$, $\chi_{ijk}(t)$, and $\chi_{ijkl}(t)$, respectively, where time t corresponds to time-step s. Similarly, let f^s_{ij}, f^s_{ijk}, and f^s_{ijkl} denote the values of $f_{ij}(x)$, $f_{ijk}(x)$, and $f_{ijkl}(x)$ at the value of x at time step s. Let h be the integration step size. For convenience, let $M^{s+1}_{im} = \left(\frac{h}{2} f^{s+1}_{im} - I_{im}\right)$. We proceed by integrating (13)-(15) of Section IV-A using trapezoidal integration:

$$
\begin{aligned}
M^{s+1}_{im}\chi^{s+1}_{mj} &= -\chi^s_{ij} - \frac{h}{2} f^s_{im}\chi^s_{mj} \\
M^{s+1}_{im}\chi^{s+1}_{mjk} &= -\chi^s_{ijk} - \frac{h}{2} f^{s+1}_{ilm}\chi^{s+1}_{mk}\chi^{s+1}_{lj} \\
&\quad - \frac{h}{2}\left[f^s_{ilm}\chi^s_{mk}\chi^s_{lj} + f^s_{il}\chi^s_{ljk}\right] \\
M^{s+1}_{im}\chi^{s+1}_{mjkl} &= -\chi^s_{ijkl} - \frac{h}{2} f^{s+1}_{inmo}\chi^{s+1}_{ol}\chi^{s+1}_{mk}\chi^{s+1}_{nj} \\
&- \frac{h}{2}\left[f^{s+1}_{inm}\chi^{s+1}_{mkl}\chi^{s+1}_{nj} + f^{s+1}_{inm}\chi^{s+1}_{mk}\chi^{s+1}_{njl} + f^{s+1}_{ino}\chi^{s+1}_{ol}\chi^{s+1}_{njk}\right] \\
&- \frac{h}{2}\left[f^s_{inmo}\chi^s_{ol}\chi^s_{mk}\chi^s_{nj} f^s_{inm}\chi^s_{mkl}\chi^s_{nj} + f^s_{inm}\chi^s_{mk}\chi^s_{njl}\right] \\
&- \frac{h}{2}\left[f^s_{ino}\chi^s_{ol}\chi^s_{njk} + f^s_{in}\chi^s_{njkl}\right].
\end{aligned}
$$

Note that these equations require multiplication of tensors (which are multilinear maps and can be thought of as higher dimensional matrices). The methods used for this multiplication will be discussed in Section V. Numerical integration proceeds by alternating between integrating the underlying dynamical system one time step, and solving the above systems of linear and multilinear equations using a linear equation solver to integrate the trajectory sensitivities one time step. To perform the integration, it is necessary to establish proper initial conditions.

C. Initial Conditions

The system considered in Section V is initially at a stable equilibrium point before being subjected to a large disturbance. We assume that $z = x^0(p)$ is an equilibrium point and C^1 function of parameter, so that $f(z) = f(x^0(p)) = 0$. Initial conditions for trajectory sensitivities of parameters are trivial. We will obtain the remaining trajectory sensitivity initial conditions by repeated differentiation of the equation $f(x^0(p)) = 0$ with respect to components of the parameter portion of the initial condition, so for $j, k, l, o > n$, and rearranging the terms:

$$
\begin{aligned}
f_{il}(z)\chi_{lj}(0) &= 0 \\
f_{il}(z)\chi_{ljk}(0) &= -f_{ilm}(z)\chi_{mk}(0)\chi_{lj}(0) \\
f_{in}(z)\chi_{njkl}(0) &= -f_{inmo}(z)\chi_{ol}(0)\chi_{mk}(0)\chi_{nj}(0) \\
&\quad - f_{inm}(z)\chi_{mkl}(0)\chi_{nj}(0) - f_{inm}(z)\chi_{mk}(0)\chi_{njl}(0) \\
&\quad - f_{ino}(z)\chi_{ol}(0)\chi_{njk}(0).
\end{aligned}
$$

Initialization is carried out by first numerically solving $f(x^0(p)) = 0$ for the initial equilibrium point $x^0(p)$, and then solving the linear and multilinear equations above using a linear equation solver to obtain the trajectory sensitivity initial conditions.

P_2	P_3	V_2	V_3	D_2	D_3	H_2	H_3	X_{12}	X_{23}
1	0.5	1	0.9	0.3	0.2	0.3	0.2	0.4	0.5

TABLE I
TEST CASE PARAMETER VALUES.

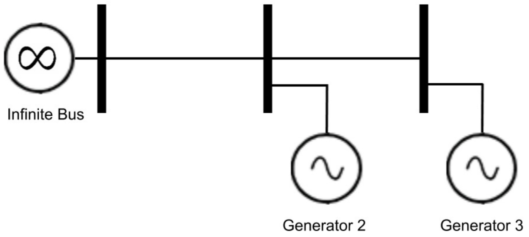

Fig. 1. Power system model used as the test case in Section V. An infinite bus, which is held at constant voltage and can draw arbitrarily high power, is connected to two synchronous generators.

V. TEST CASE

The test case considered is a simple model of a power system which consists of two synchronous generators connected to two buses arranged radially, with an infinite bus at the head of the feeder, as shown in Fig. 1. The infinite bus is held at constant voltage of 1 per unit (p.u.) and can draw arbitrarily high power to match the power production of the two generators. Bus 1 is the infinite bus, generator 2 is connected to bus 2, and generator 3 to bus 3. The rotor angle and angular velocity for generator 2 are given by δ_2 and ω_2, respectively, and δ_3 and ω_3 similarly for generator 3. Let $y = [\delta_2\ \omega_2\ \delta_3\ \omega_3]^\intercal$ be the vector of state variables. Let P_2, D_2, H_2, and V_2 be the mechanical power, damping coefficient, moment of inertia, and voltage magnitude at generator 2, and define P_3, D_3, H_3, and V_3 analogously for generator 3. Let X_{ij} denote the impedance on the line from bus i to bus j. Parameter values used are given in Table I. Let $p = [P_2\ P_3\ V_2\ V_3\ D_2\ D_3\ H_2\ H_3]^\intercal$ be the vector of chosen parameters, and let $x = [y^\intercal\ p^\intercal]^\intercal$. Then the dynamics of this system are given by:

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
H_2\dot{x}_2 &= P_2 - D_2 x_2 - \sin(x_1)\frac{V_2}{X_{12}} \\
&\qquad - \left(\sin(x_1)\cos(x_3) - \cos(x_1)\sin(x_3)\right)\frac{V_2V_3}{X_{23}} \\
\dot{x}_3 &= x_4 \\
H_3\dot{x}_4 &= P_3 - D_3 x_4 \\
&\qquad - \left(\sin(x_3)\cos(x_1) - \cos(x_3)\sin(x_1)\right)\frac{V_2V_3}{X_{23}},
\end{aligned}
$$

which can be written more succinctly as

$$\dot{x} = f(x). \tag{16}$$

Let p_0 be the parameter values given in Table I. There exists a stable equilibrium point of (16) at $y^0(p_0) =$

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

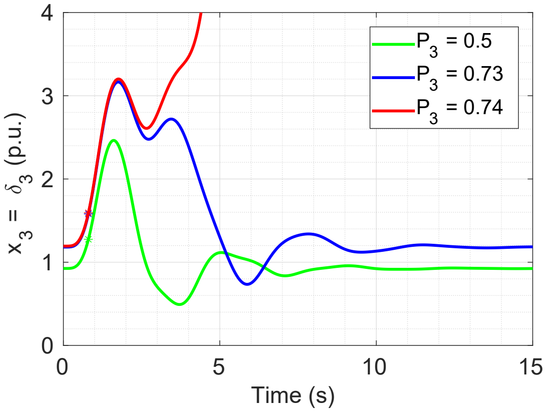

Fig. 2. For the test case of Section V, $x_3 = \delta_3$ is shown as a function of time for (a) the initial value $P_3 = 0.5$ (green), (b) the critical value $P_3 = 0.73$ (blue), and (c) just over the critical value at $P_3 = 0.74$ (red). Disturbance occurs at $t = 0$ and stars indicate when it is cleared.

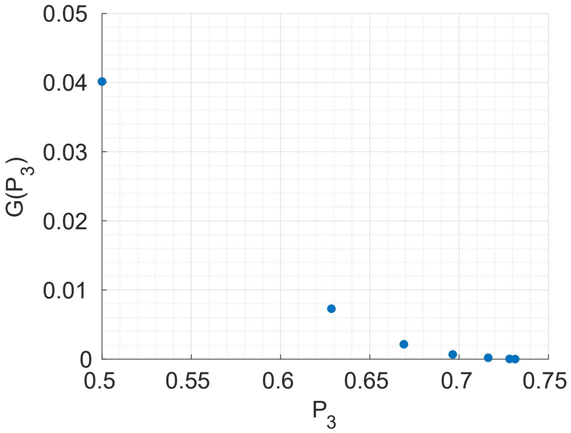

Fig. 3. Computation of critical parameter value for P_3 using the one-dimensional parameter space algorithm of Section III-A. Iterations begin in the top left and proceed towards the bottom right. Rapid convergence to the critical parameter value is observed.

$[0.6435\ 0\ 0.9250\ 0]^\intercal$. For p reasonably near p_0 there exists a stable equilibrium point $y^0(p)$ of (16) close to $y^0(p_0)$ which can be found, for example, by solving $f(y^0(p)) = 0$ using Newton-Raphson with initial condition p_0. The system is initially at the stable equilibrium point $x^0(p)$, then the line between buses 1 and 2 trips out of service for a duration of 0.8 s. The disturbance is modeled by setting the term $\sin(x_1)\frac{V_2}{X_{12}}$ to zero in equation f_2, leaving the other components of f unchanged. Before and after the disturbance, the system dynamics are given by (16).

Fig. 2(a) shows $x_3 = \delta_3$ as a function of time for $p = p_0$. Note that the system is initially at a stable equilibrium point. Then the disturbance occurs and the angle δ_3 deviates far from equilibrium. After 0.8 s the disturbance is cleared, normal system dynamics are restored, and the system evolves until δ_3 returns to its prior equilibrium value. This picture is typical for cases in which the system is able to recover from the fault.

In order to apply the algorithms of Section III, numerical computation of the trajectory sensitivities as described in Section IV is necessary. This computation required tensor multiplication operations and solutions of multilinear systems of equations. These were performed by reshaping higher dimensional tensors into matrices, performing matrix multiplication or solving a linear system of equations, and then reshaping back into higher dimensions. Such techniques work due to multilinearity of tensors.

A. One-Dimensional Parameter Space

For each $p_i \in p$, the algorithm detailed in Section III-A was applied to find the nearest critical parameter value of p_i. There were some parameters, namely H_2, H_3, and D_3, where varying just one at a time was not sufficient to cause the system to fail to recover from the fault (for positive values of the parameters, as is required physically). In these cases, the algorithm attempted to send the parameter values towards zero or negative, so it was quickly clear that they were unable to induce system non-recovery.

Among the remaining parameters, $p = P_3$ is representative of the observed behavior of the algorithm. Fig. 3 shows the convergence of the one-dimensional parameter space algorithm for $p = P_3$. Although the critical value of P_3 is about 50% larger than its initial value, convergence to its critical value is rapid and monotonic. At the final iterate, $G(P_3) < 10^{-5}$. Fig. 2 shows that the trajectory corresponding to the critical parameter value of $P_3 = 0.73$ p.u. is in fact marginally stable. More generally, for each $p_i \in p$ (other than those mentioned above which could not drive the system to non-recovery), rapid, monotonic convergence to the corresponding critical value was observed. Each iteration of the algorithm took approximately 2 s, and the algorithm converged in $6-10$ iterations for each case, for a total runtime of $12-20$ s.

B. Multi-Dimensional Parameter Space

Several choices of multi-dimensional parameter spaces were tested. The weighting matrix A was set to the identity matrix in all cases. The initial parameter values p_0 were taken from Table I. First, the choice $p = p_i$ for each i was made to confirm that the multi-dimensional parameter space algorithm of Section III-B gave the same critical parameter values as the one-dimensional parameter space algorithm of Section III-A. Next, the following three sets of parameters were considered:

$$S_4 := \{P_2, P_3, V_2, V_3\}$$
$$S_6 := \{P_2, P_3, V_2, V_3, D_2, D_3\}$$
$$S_8 := \{P_2, P_3, V_2, V_3, D_2, D_3, H_2, H_3\}.$$

Note that S_4 consists of all the parameter values that power system operators can select during real-time operation, S_6 consists of S_4 together with the parameter values that can be tuned by control engineers (namely, the damping that is a result of controller design), and S_8 consists of S_6 together with the remaining generator parameter values.

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.

Parameter Set	Objective Function Value
S_4	0.0075
S_6	0.0062
S_8	0.0053

TABLE II

VALUES OF THE OBJECTIVE FUNCTION (4) FOR THE CRITICAL PARAMETER VALUES OBTAINED BY THE MULTI-DIMENSIONAL OPTIMIZATION ALGORITHM FOR THE PARAMETER SETS S_4, S_6, AND S_8.

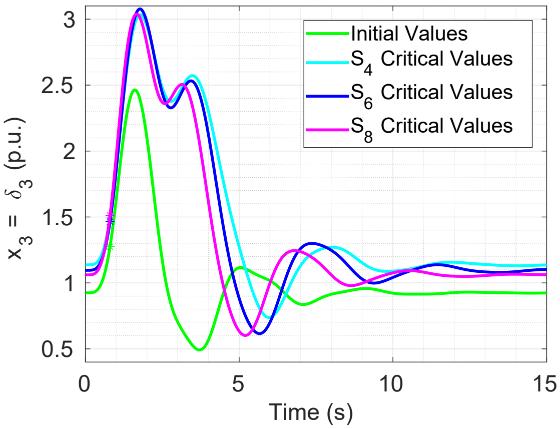

Fig. 4. The state $x_3 = \delta_3$ is shown as a function of time for (a) the initial parameter values (green), (b) the critical values for S_4 (cyan), (c) the critical values for S_6 (blue), and (d) the critical values for S_8 (magenta). Disturbance occurs at $t = 0$ and stars indicate when it is cleared.

We chose $\epsilon = 10^{-5}$ for the constraint (5). Then, the multi-dimensional parameter space algorithm converged for S_4 in 9 iterations, for S_6 in 9 iterations, and for S_8 in 8 iterations. At about 2 s per iteration, runtimes varied over 15 – 20 s. Let p_4^*, p_6^*, and p_8^* denote the critical parameter values obtained via this algorithm for S_4, S_6, and S_8, respectively. Then

$$\begin{aligned}
p_4^* &= \begin{bmatrix}1.0657 & 0.5651 & 0.9198 & 0.8986\end{bmatrix}^\mathsf{T}\\
p_6^*(1:3) &= \begin{bmatrix}1.0555 & 0.5549 & 0.9352\end{bmatrix}^\mathsf{T}\\
p_6^*(4:6) &= \begin{bmatrix}0.8991 & 0.2618 & 0.1736\end{bmatrix}^\mathsf{T}\\
p_8^*(1:4) &= \begin{bmatrix}1.0466 & 0.5462 & 0.9489 & 0.8992\end{bmatrix}^\mathsf{T}\\
p_8^*(5:8) &= \begin{bmatrix}0.2648 & 0.1751 & 0.2767 & 0.1647\end{bmatrix}^\mathsf{T}.
\end{aligned}$$

The parameter values satisfy $G(p_4^*) = G(p_6^*) = G(p_8^*) = \epsilon$, so they are all critical values. Furthermore, Fig. 4 shows that the trajectories corresponding to these critical values are marginally stable.

The goal was to obtain critical parameter values which minimized the objection function (4). Table II shows the values of the objective functions at p_4^*, p_6^* and p_8^*. As additional parameters are introduced, more options for critical parameter values become available, some of which may be closer to p_0 than the previous options. Consequently, the objective function value, which measures the distance of the chosen critical parameter value from p_0, decreased from S_4 to S_6, and again from S_6 to S_8. Overall, for all the chosen combinations of parameters, the multi-dimensional parameter space algorithm converged rapidly to a critical parameter value which appears to be the closest critical parameter value to p_0.

VI. CONCLUSION

Novel algorithms for efficient numerical computation of critical parameter values have been developed. In particular, the critical parameter values nearest to a specified set of parameters were computed for both one and arbitrary parameter space dimensions. Unlike prior work, these algorithms do not require the existence of energy functions, do not assume parameter independent post-disturbance dynamics, and do not require identification of a particular unstable equilibrium point in state space. Consequently, they represent practical algorithms for critical parameter value computation. Computation of third-order trajectory sensitivities was derived for the algorithms. The algorithms were successfully validated on a simple power system model. Future work will involve extending the algorithms and theory to systems of differential algebraic equations.

REFERENCES

[1] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Stipes Publishing Co., 2007.

[2] H.-D. Chiang and L. F. C. Alberto, *Stability Regions of Nonlinear Dynamical Systems: Theory, Estimation, and Applications*. Cambridge University Press, 2015.

[3] H.-D. Chiang, *Direct Methods for Stability Analysis of Electric Power Systems*. John Wiley & Sons, Inc., 2011.

[4] R. T. Treinen, V. Vittal, and W. Kliemann, "An improved technique to determine the controlling unstable equilibrium point in a power system," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 43, no. 4, pp. 313–323, 1996.

[5] M. W. Fisher and I. A. Hiskens, "Numerical computation of parameter-space stability/instability partitions for induction motor stalling," *International Federation of Automatic Control - PapersOnLine*, vol. 49, no. 27, pp. 250–255, 2016.

[6] M. W. Fisher and I. A. Hiskens, "Numerical computation of critical parameter values for fault recovery in power systems," in *Proc. Power Systems Computation Conference*, 2018.

[7] M. W. Fisher and I. A. Hiskens, "Parametric dependence of large disturbance response and relationship to stability boundary," in *Proc. IEEE 56th Conference on Decision and Control*, pp. 1821–1827, 2017.

[8] M. W. Hirsch, C. C. Pugh, and M. Shub, *Invariant Manifolds*, vol. 583 of *Lecture Notes in Mathematics*. Springer-Verlag, 1977.

[9] J. Palis, "On Morse-Smale dynamical systems," *Topology*, vol. 8, no. 4, pp. 385–405, 1969.

[10] I. A. Hiskens and M. A. Pai, "Trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems*, vol. 47, no. 2, pp. 204–220, 2000.

[11] S. Geng and I. A. Hiskens, "Second-order trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems I*, vol. 66, no. 5, pp. 1922–1934, 2019.

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 06,2020 at 02:46:40 UTC from IEEE Xplore. Restrictions apply.