Determining Parameter Space Margins for Fault Recovery By Inverse Sensitivity Minimization

Michael W. Fisher   Ian A. Hiskens
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan

***Abstract*—Power system model parameter values are becoming increasingly uncertain and time-varying. Therefore, it is important to determine the margin in parameter space between a given set of parameter values for which the system will recover from a particular fault, and the nearest parameter values for which it will not recover from that fault. This work presents an efficient method for computing parameter space recovery margins by exploiting the property that the trajectory becomes infinitely sensitive to small changes in parameter value along the operating point's region of attraction boundary. Consequently, along this boundary the inverse sensitivity of the trajectory approaches zero. The method proceeds by varying parameter values so as to minimize the inverse sensitivity of the system trajectory. Recent results provide theoretical justification for the approach. The efficacy of the method is demonstrated using a modified IEEE 39-bus New England power system test case.**

***Index Terms*—power system dynamics, large disturbance stability.**

I. Introduction

Faults are inevitable in power system operation. Whether the system recovers from a particular fault to a desired operating point depends upon many factors, such as loading pattern, controller set-points and load characteristics. In studying such events, it is an important though challenging problem to identify, for a particular fault, the set of parameter values for which the system will recover from that fault to a given operating point. We term this region of parameter space the *recovery region* (RR) corresponding to that fault. For a particular fault, its *recovery region boundary* is the boundary of the RR in parameter space, and its *recovery margin* is the distance from a given nominal set of parameter values to the RR boundary.

Typical industry practice involves the construction of inner approximations of the RR boundary known as nomograms in two or three dimensional parameter space using hyperplanes [1]. Several extensions to these traditional techniques, such as [2], have been proposed. However, the number of computation-intensive dynamic simulations required by the methods increases rapidly with parameter dimension. Consequently, they are still limited to low dimensional parameter spaces and are constructed in an off-line process rather than being determined in a real-time operational context. As the small number of parameters of interest are chosen based on operator intuition and experience, many parameters which could be critical to system recovery may therefore be overlooked. Furthermore, the methods are inherently approximate and can be overly conservative. Hence, it is desirable to develop techniques for more accurate and efficient computation of the RR boundary in lower parameter dimensions, and for rapid assessment of recovery margins in higher dimensional parameter space.

Much of the classical work in rapid assessment of recovery margins has involved the use of energy functions to estimate the region of attraction (RoA) of the desired operating point [3], [4], [5], [6]. However, many industry standard power system models are not amenable to the development of energy functions and, even when such functions exist, their RoA estimates tend to be overly conservative in practice. Furthermore, these methods require knowledge of a particular equilibrium point, called the controlling unstable equilibrium point (CUEP), with specific dynamical properties. Locating such points is computationally intractable for realistic power systems. Finally, the methods are restricted to parameters which do not influence post-fault dynamics, such as the fault clearing time, which excludes many important system parameters.

Recently, efficient methods were developed to numerically compute the RR boundary to arbitrary precision in two dimensional parameter space, and to determine the recovery margins in arbitrary dimensional parameter space, without the use of energy functions and for parameters which influence post-fault dynamics [7], [8]. However, they still require prior knowledge of the CUEP. This paper improves upon that recent work by presenting techniques which do not require knowledge of the CUEP.

The key idea is that, for an initial condition which lies exactly on the RoA boundary, its trajectory is infinitely sensitive to small perturbations; incremental changes in initial condition could push the trajectory either inside or outside the RoA. As small changes in parameter values cause perturbations to the post-fault initial condition and the RoA boundary, the trajectory corresponding to a parameter value on the RR boundary becomes infinitely sensitive to small changes in parameter values. Algorithms are introduced to compute parameter values on the RR boundary by varying parameters so as to maximize trajectory sensitivities. For implementation, however, it is advantageous to minimize the inverse sensitivities.

The authors gratefully acknowledge the contribution of the U.S. National Science Foundation through grant ECCS-1810144.

These ideas are applied to find the nearest point on the RR boundary for a specified direction in parameter space, and to numerically trace the RR boundary for two dimensional parameter space. These developments also motivate algorithms for computing recovery margins in arbitrary dimensional parameter space. For clarity of presentation, limits on AVR and PSS controller states are not included in this work, but these limits can be incorporated into the algorithms by computing trajectory sensitivities for hybrid systems as in [9]. The theory justifying the algorithms has recently been extended to incorporate discrete events, such as controller limits [10].

The paper is organized as follows. Section II provides the theoretical setting and justification for the algorithms, with the details of the algorithms given in Section III. Section IV introduces the IEEE 39-bus New England power system test case, and Section V demonstrates the algorithm characteristics. Section VI offers some concluding remarks.

II. Theory

Power systems are commonly modeled as a system of differential and algebraic equations (DAE):

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

$$0 = g(x, y) \tag{2}$$

where $x \in \mathbb{R}^n$ represents dynamic states such as rotor angles and frequencies, $y \in \mathbb{R}^m$ represents algebraic states such as voltages and currents, $f : \mathbb{R}^{n+m} \to \mathbb{R}^n$, and $g : \mathbb{R}^{n+m} \to \mathbb{R}^m$. For simplicity of presentation, in the subsequent discussion we will assume that f and g are continuously differentiable, but recent theoretical work [10] has shown that the following results extend to a large class of hybrid systems which permit restricted switching behavior, such as controller limits, provided grazing of switching surfaces does not occur. Such hybrid dynamical systems more accurately reflect realistic power system behavior.

The desired steady state operating point is given by a stable equilibrium point (SEP) of (1)-(2) which has an associated RoA. Consider a particular fault, and let its post-fault initial condition be the system state at the instant when the fault clears. Let $\mathbb{R}^P$ be a space of parameters of the system, and let $p \in \mathbb{R}^P$ denote a set of parameter values. As the parameter value p varies, so do the RoA and the post-fault initial condition. Let $x(p, z_0, t)$ denote the dynamic states in the solution of (1)-(2) for parameter value p from initial condition z_0 for time t. Let $z(p)$, with $z : \mathbb{R}^P \to \mathbb{R}^n$, denote the post-fault initial condition corresponding to parameter value p.

Trajectory sensitivities are partial derivatives of the states with respect to initial conditions or parameters. They can be efficiently computed as a byproduct of the numerical integration of the underlying dynamics [9], [11]. For any parameter p, define the trajectory sensitivity χ (of the dynamic states) with respect to p as:

$$\chi(p, t) := \frac{\partial x(p, z(p), t)}{\partial p}. \tag{3}$$

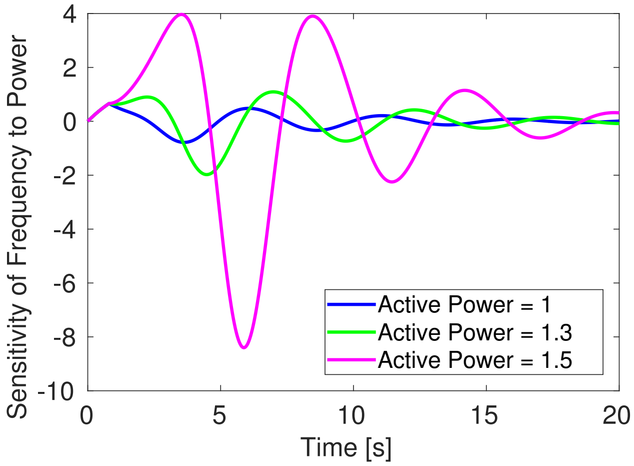

Fig. 1. The sensitivity of the frequency of a classical machine to generator mechanical power as a function of time, as the generator power approaches the RR boundary.

This is the partial derivative of the dynamic states of the system at time t starting from post-fault initial condition $z(p)$. These trajectory sensitivities measure the sensitivity of the system dynamic states to small changes in parameter value.

To develop intuition for the behavior of trajectory sensitivities for parameter values near the RR boundary, consider the simple example of a single machine infinite bus with classical machine dynamics. A fault in this system is modeled by setting the electrical torque to zero until the fault is cleared. Fig. 1 shows the sensitivity of the frequency to generator mechanical power as that power value approaches the RR boundary. Observe that the sensitivities grow larger and larger as the RR boundary is approached. More generally, the intuition behind this is that at the RR boundary the trajectory becomes infinitely sensitive to small changes in parameter values, because small perturbations in parameter values could push it to either side of the boundary of the RoA - causing it either to recover from the fault or not.

To generalize the above intuition, define the scalar function $G : \mathbb{R}^P \to \mathbb{R}$ by:

$$H(p, t) = \frac{1}{||\chi(p, t)||_1}, \tag{4}$$

$$G(p) = \inf_{t \geq 0} H(p, t), \tag{5}$$

where the matrix norm $||M||_1 = \sum_{i,j} |M_{ij}|$ for any matrix M. So, $G(p)$ represents the minimum over time of the inverse of the norm of the trajectory sensitivities. Hence, trajectory sensitivities diverging to infinity is equivalent to G approaching zero. Under practical assumptions, for a large class of nonlinear systems the following result has been proven in [12]: G is well-defined, continuous, non-negative, and the RR boundary is equal to $G^{-1}(0)$. In particular, this implies that solving for p such that $G(p) = 0$ will drive p onto the RR boundary. This motivates the development of algorithms for computing the RR boundary.

III. Algorithm

Based on the theory of Section II, the RR boundary consists precisely of the parameter values p such that $G(p) = 0$. This section formulates algorithms for finding the closest point on the RR boundary in the case of one dimensional parameter space, and for numerically tracing the RR boundary in two dimensional parameter space.

A. One Dimensional Parameter Space

To find the closest point on the RR boundary, the goal is to solve

$$G(p) = 0. \tag{6}$$

Earlier theoretical work [13] and numerical experiments [14] suggest that G is approximately affine for parameter values near the RR boundary. Motivated by this linear structure, we solve (6) using Newton-Raphson with backtracking, which is an iterative algorithm that converges rapidly for affine equations. At each iteration s, we perform a full time-domain simulation to evaluate G and its derivative DG. The results of that simulation also reveal whether p^s lies inside or outside the RR, based on whether the system recovers or does not recover to the desired SEP. Let k be the largest iteration with $k \leq s$ such that p^k lies in the RR. In other words, k represents the most recent iteration at which p^k lies in the RR. Let $\kappa = \left(\frac{1}{2}\right)^{s-k}$ denote the step length. We perform the update

$$p^{s+1} = p^k - \kappa \; G(p^k)/DG(p^k). \tag{7}$$

If p^s lies inside the RR, so that $k = s$, then $\kappa = 1$ and this reduces to the standard Newton-Raphson update. On the other hand, if p^s lies outside the RR, then we start from the most recent value p^k that lies inside the RR, determine the direction of its Newton-Raphson update, and do a backtracking line search along that direction until a parameter value inside the RR is reached. In particular, the backtracking is performed by reducing the distance traveled along the Newton-Raphson search direction by a factor of two at every iteration until the RR is encountered.

Performing the update of (7) requires computation of the derivative DG. For any parameter value p that lies inside the RR, there exists a time $\hat{t}(p)$ such that

$$G(p) = \min_{t \geq 0} H(p,t) = H(p, \hat{t}(p)). \tag{8}$$

Since $\hat{t}(p)$ is the point at which $H(p,t)$ achieves a minimum in time, it is an extremal point of $H(p,t)$, which implies that

$$\frac{\partial H(p, \hat{t}(p))}{\partial t} = 0. \tag{9}$$

We compute

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

where the last step follows by substituting in (9). So, to compute DG it suffices to compute $\frac{\partial H}{\partial p}$. This is given by

$$\frac{\partial H}{\partial p}(p,t) = \frac{-\frac{\partial \chi(p,t)}{\partial p} \cdot \mathrm{sign}(\chi(p,t))}{||\chi(p,t)||_1^2} \tag{10}$$

where for any vectors v and w, $(\mathrm{sign}(v))_i = 1$ if $v_i \geq 0$, $(\mathrm{sign}(v))_i = -1$ if $v_i < 0$, and $v \cdot w = \sum_i v_i w_i$ denotes the standard Euclidean dot product. Note that (10) includes $\frac{\partial \chi(p,t)}{\partial p}$ which is a second order trajectory sensitivity and can be obtained efficiently as a byproduct of the underlying integration [11], [12].

Computation of G and its derivative DG proceeds as follows. During a time-domain simulation corresponding to parameter value p^s, the time $\hat{t}(p^s)$ at which $H(p^s, t)$ achieves its minimum is observed. Then $G(p^s) = H(p^s, \hat{t}(p^s))$, and $DG(p^s) = \frac{\partial H}{\partial p}(p^s, \hat{t}(p^s))$ is computed using (10). These are then used to perform the update of (7) and compute p^{s+1}. This process is repeated iteratively until $G(p^s)$ converges to zero, which causes p^s to converge to the RR boundary.

B. Two Dimensional Parameter Space

By the results of Section II, the RR boundary is equal to $G^{-1}(0)$. In two dimensional parameter space, $G : \mathbb{R}^2 \to \mathbb{R}$ is one equation with two free variables (parameters), so the RR boundary is typically a one dimensional curve because there is one more free variable than equations. The goal is to numerically trace this curve by iteratively computing a sequence of points along the curve. As numerical precision is finite, instead of tracing the curve $G^{-1}(0)$, we trace the curve $G^{-1}(\epsilon)$ for some small $\epsilon > 0$. This approximates the true RR boundary to arbitrary precision, with the tolerance level set by ϵ. Tracing the curve $G^{-1}(\epsilon)$ is accomplished using the following continuation method, which alternates between a predictor step and a corrector step, as shown in Fig. 2. Let s denote the current iteration, and let p^s be the current parameter values on the curve. Then p^s must satisfy $G(p^s) = \epsilon$.

The predictor step generates a first order prediction of the next point on the curve, and proceeds as follows. First, we obtain the unit tangent vector to the curve $G^{-1}(\epsilon)$ at p^s. This can be done by noting that $DG(p^s)$ is orthogonal to the curve $G^{-1}(\epsilon)$, because G is constant along this curve so its derivative DG must be orthogonal to it. Then, any vector orthogonal to $DG(p^s)$ must be tangent to $G^{-1}(\epsilon)$ since the parameter space

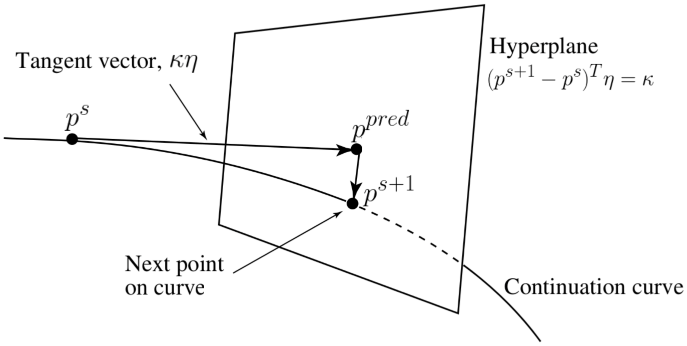

Fig. 2. The predictor-corrector continuation process for iteratively tracing a sequence of points along a curve.

is exactly two dimensional. We can compute DG by noting that,

$$DG(p) = \begin{bmatrix}\frac{\partial G}{\partial p_1}(p) & \frac{\partial G}{\partial p_2}(p)\end{bmatrix} \tag{11}$$

$$\frac{\partial G}{\partial p_i}(p) = \frac{\partial H}{\partial p_i}(p, \hat{t}(p)), \quad \text{for } i \in \{1,2\}, \tag{12}$$

where the latter equation follows since $G(p) = H(p, \hat{t}(p))$ for any p inside the RR, by the argument in Section III-A. Define the following notation for $i, j \in \{1, 2\}$:

$$\chi_i(p,t) := \frac{\partial x(p, z(p), t)}{\partial p_i}$$

$$\chi_{ij}(p,t) := \frac{\partial^2 x(p, z(p), t)}{\partial p_i \partial p_j}.$$

By an analogous argument to that in Section III-A, for $i \in \{1, 2\}$ we have

$$\frac{\partial H}{\partial p_i}(p,t) = \frac{-\chi_{1i}(p,t)\cdot \text{sign}(\chi_1(p,t))}{||\chi(p,t)||_1^2} - \frac{\chi_{2i}(p,t)\cdot \text{sign}(\chi_2(p,t))}{||\chi(p,t)||_1^2}.$$

This formula can be used to compute $DG(p^s)$ at each iteration. Then, writing $DG(p^s) = [a\ b]$, a tangent vector to the curve is given by $[b\ \ {-a}]^T$. Let η denote $[b\ \ {-a}]^T$ divided by its norm, so that η is the unit tangent vector to the curve. Let $\kappa \in \mathbb{R}$ be the step size for the prediction. The predicted point is given by

$$p^{pred} = p^s + \kappa\eta. \tag{13}$$

Once a predicted point has been obtained, the next step is to correct it back onto the curve $G^{-1}(\epsilon)$. To do so, an orthogonal projection from the predicted point onto the curve is performed. In particular, the next point on the curve p^{s+1} is chosen such that the vector $(p^{s+1} - p^{pred})$ is orthogonal to the vector $(p^{pred} - p^s)$, so that

$$(p^{s+1} - p^{pred})^T(p^{pred} - p^s) = 0.$$

Using (13) and the fact that η is a unit vector so $\eta^T\eta = 1$, the above constraint simplifies to

$$(p^{s+1} - p^s)^T\eta - \kappa = 0. \tag{14}$$

So, correction back onto the curve requires solution of the system of equations:

$$F(p^{s+1}) := \begin{bmatrix} G(p^{s+1}) - \epsilon \\ (p^{s+1} - p^s)^T\eta - \kappa \end{bmatrix} = 0. \tag{15}$$

To solve $F(p^{s+1}) = 0$, which consists of two equations with two free variables (parameters), the standard Newton-Raphson algorithm is applied. This requires the derivative of F, which is given by

$$DF(p^{s+1}) = \begin{bmatrix} DG(p^{s+1}) \\ \eta^T \end{bmatrix}.$$

Repeating iterations of Newton-Raphson will find p^{s+1} such that $F(p^{s+1}) = 0$, which ensures that $G(p^{s+1}) = \epsilon$, so p^{s+1} is on the curve $G^{-1}(\epsilon)$. The prediction and correction steps are then alternated iteratively until the RR boundary has been traced.

IV. Model

The IEEE 39-bus New England power system test case shown in Fig. 3 will be used to illustrate the algorithms of Section III. Generators are modeled using a 4th order machine model as given in [3], AVR and PSS models are based on the IEEE standard models PSS 1A and Exciter ST1 [15], and the full set of dynamic equations and system parameters are given in [16]. As the theory and algorithms above are presented for smooth systems, the test case is modified to remove the limits on AVR and PSS controller states. However, future work aims to reintroduce these limits, and recent theoretical developments [10] suggest that these algorithms should be successful with the controller limits included as well. A fault occurs at bus 16, and is cleared after $0.2s$. The fault is modeled as a switched, constant shunt reactance with $X_{fault} = 0.001$p.u.

Many model parameters of the system are of interest for recovery considerations. A background load scaling factor (SF) is introduced which multiplies the active and reactive power background loads at every bus in the network. As background load is time-varying and uncertain, it is a natural choice for assessing system recoverability. An AVR gain SF multiplies the AVR gain for every generator, and helps to capture the impact of controller tuning on system stability. The load active and reactive power are represented by the standard exponential voltage load model. The voltage exponents are set equal for all background loads. As load dynamics are notoriously difficult to model, this parameter serves to quantize the impact of uncertain load behavior on system recoverability.

V. Results

The algorithms of Section III were demonstrated on the test case of Section IV. The dynamic states appearing in the first and second order trajectory sensitivities used in Section III were restricted to the generator dynamic states to avoid the possibility for the sensitivities of internal controller states to overshadow the sensitivities of the physical generator states. The algorithm for one dimensional parameter space, which finds the closest point on the RR boundary to an

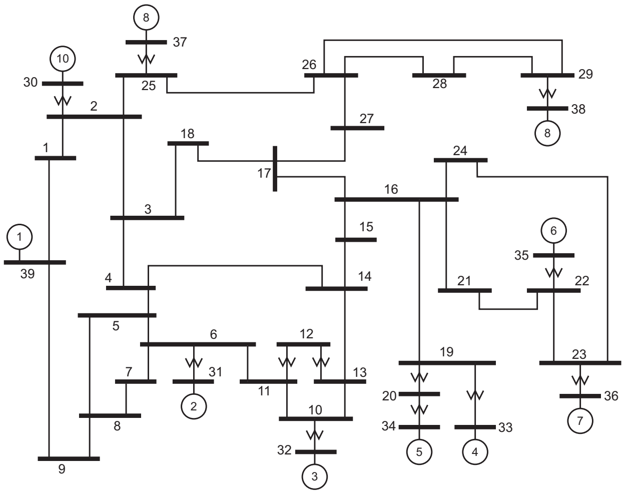

Fig. 3. IEEE 39-bus New England power system.

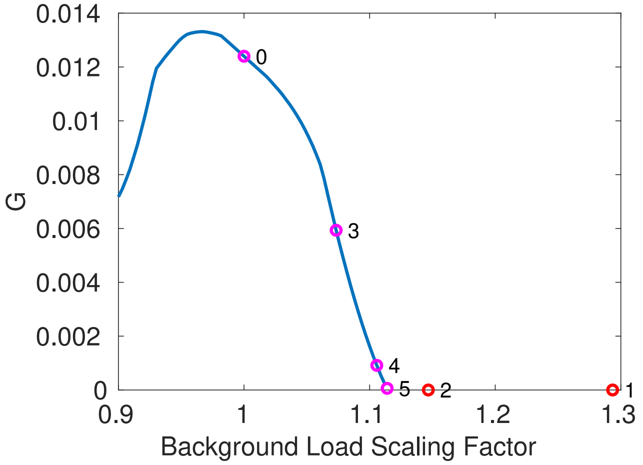

Fig. 4. G as a function of background load scaling factor (blue line). The iterations of the one dimensional parameter space algorithm are labeled in order (circles), including both those that lie outside (red) and inside (magenta) the RR.

initial given parameter value, was applied with the parameter of interest being the background load SF. Fig. 4 illustrates that the algorithm converged to the RR boundary in just 5 iterations (6 simulations total). Furthermore, it shows that G is approximately linear near the RR boundary, as predicted by the theory in Section II. Hence, this algorithm rapidly and accurately determines the nearest point on the RR boundary in one dimensional parameter space.

To observe the approach to instability, note that under stressed conditions the first generator to go unstable is generator 2. Fig. 5 shows the frequency of generator 2 as a function of time for all values of the background load SF which lie inside the RR that were attained during the algorithm of Fig. 4. Observe that as the background load SF approaches the RR boundary, the fluctuations in the frequency of generator 2 grow larger, indicative of proximity to instability.

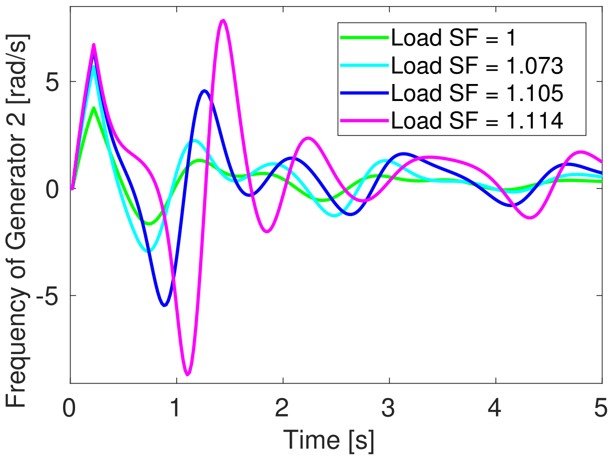

Fig. 5. Frequency of generator 2 as a function of time for the values of the background load scaling factor that correspond to the iterations (inside the RR) of the one dimensional algorithm.

Once a point on the RR boundary has been identified, the continuation method described in Section III can be applied to numerically trace the RR boundary in two dimensional parameter space. This algorithm was applied for the two dimensional parameter space consisting of the AVR gain SF and the load voltage exponent. The tolerance was set to $\epsilon = 10^{-5}$. Fig. 6 shows the RR boundary and the RR in this parameter space. If the AVR gains are set about 5% less than nominal (SF=1.0), the system will not recover regardless of the load exponent. For slightly higher AVR gain SFs, a nonempty RR emerges. In this regime of the AVR gain SFs, if the load voltage exponents are sufficiently low then the parameter values will lie outside the RR. In other words, the system becomes unable to recover as the loads approach constant power loads. This is consistent with standard intuition that constant power loads have a more detrimental impact on system stability than constant impedance loads. However, if load voltage exponents are sufficiently high then the parameter values will also lie outside the RR. Hence, there are two ranges of load exponents outside the RR, corresponding to loads which approach constant power, and also corresponding to loads which approach constant impedance. The latter behavior is counterintuitive, and serves as an example of how these algorithms have the potential to reveal unexpected dynamic behaviors which would not have been observed otherwise.

To observe the influence of variations in load voltage exponents on system recovery, recall that generator 2 is the first generator to go unstable as a result of the fault. Fig. 7 shows the relative angle of generator 2 as a function of time for a fixed AVR gain SF and for several load voltage exponent values. For sufficiently high or sufficiently low exponents, the system goes unstable and is unable to recover from the fault. For exponents near the top and bottom of the RR of Fig. 6, angle fluctuations are larger and occur later in time. For exponent values between these upper and lower RR boundary exponents, the angle shows smaller fluctuations with a peak occurring earlier in time. Hence, for high or low load voltage

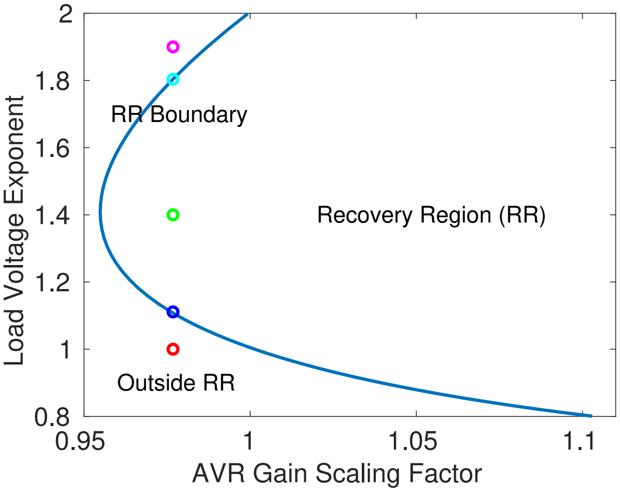

Fig. 6. The RR boundary and the RR in the two dimensional parameter space of AVR gain SF and load voltage exponent. Colored circles indicate the parameter values whose corresponding dynamic behavior is shown in Fig. 7.

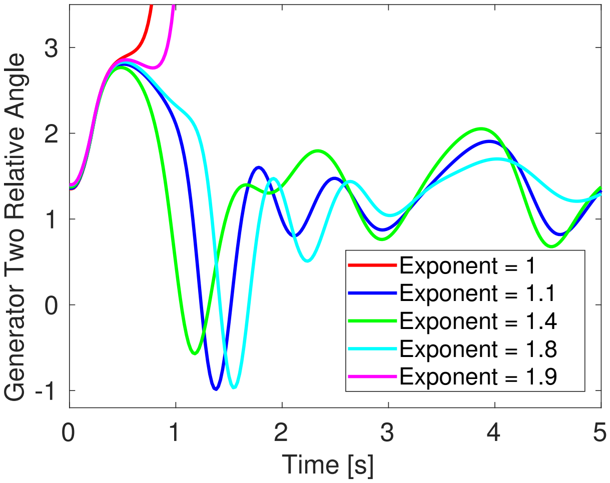

Fig. 7. The angle of generator two relative to the angle of generator one as a function of time for an AVR gain SF of 0.9768 and a range of load voltage dynamics exponents.

exponents, angle fluctuations grow and lead to instability, whereas for an intermediary range of exponents, the angles are able to recover from the transient disturbance. Overall, the algorithms were successfully applied to the modified IEEE 39-bus power system to find the closest point on the RR boundary in one dimensional parameter space, and to numerically trace the RR boundary in two dimensional parameter space.

VI. Conclusion

This work developed algorithms to find the closest point on the RR boundary in one dimensional parameter space, and to numerically trace the RR boundary in two dimensional parameter space. Unlike current industry practice, these algorithms are not approximate or conservative, and are able to efficiently compute the RR boundary to arbitrary precision. Furthermore, the proposed algorithms can handle parameter dependent post-fault dynamics, are generalizable to hybrid systems, and do not require prior knowledge of the controlling unstable equilibrium point.

The algorithms are demonstrated on the modified IEEE 39-bus New England power system, and reveal an unexpected negative influence of constant impedance loads on system recovery that would not have been identified otherwise.

Extending these algorithms to the computation of recovery margins in arbitrary dimensional parameter space, similar to the approach in [12], is currently underway. This will provide a tool for identifying parameters that are critical to system stability but may have been overlooked, as well as a quantitative metric for recovery assessment. Future work will explicitly illustrate the algorithms with the AVR and PSS controller limits included, and will overlay the RR boundaries for multiple faults.

References

[1] J. D. McCalley, S. Wang, R. T. Treinen, and A. D. Papalexopoulos, “Security boundary visualization for systems operation,” *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 940–947, 1997.

[2] Y. V. Makarov, P. Du, S. Lu, T. B. Nguyen, X. Guo, J. W. Burns, J. F. Gronquist, and M. A. Pai, “PMU-based wide-area security assessment: Concept, method, and implementation,” *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1325–1332, 2012.

[3] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*, 1997.

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

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

[6] 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.

[7] 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.

[8] ——, “Numerical computation of critical parameter values for fault recovery in power systems,” *2018 Power Systems Computation Conference (PSCC)*, pp. 1–6, 2018.

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

[10] M. W. Fisher and I. A. Hiskens, “Parametric dependence of large disturbance response for vector fields with event-selected discontinuities,” *18th European Control Conference (ECC)*, pp. 166–173, 2019.

[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.

[12] M. W. Fisher and I. A. Hiskens, “Numerical computation of critical system recovery parameter values by trajectory sensitivity maximization,” *58th Conference on Decision and Control (CDC)*, 2019, to appear.

[13] S. Roy and I. Hiskens, “Inverse-affine dependence of recovery-time sensitivities on critical disturbance parameters: A nonlinear dynamics explanation,” *American Control Conference (ACC)*, pp. 4452–4456, 2012.

[14] T. B. Nguyen, M. A. Pai, and I. A. Hiskens, “Computation of critical values of parameters in power systems using trajectory sensitivities,” *14th Power Systems Computation Conference (PSCC)*, pp. 1–6, 2002.

[15] IEEE Standard 421.5, Tech. Rep., 1982.

[16] Benchmark Systems for Stability Controls Task Force, “Benchmark systems for small-signal stability analysis and control,” IEEE Power and Energy Society, Tech. Rep., 2015.