Approximating a post-contingency stable operation region in parameter space through time-domain simulation

Conference Paper · August 2013

DOI: 10.1109/IREP.2013.6629361

CITATION
1

READS
48

3 authors, including:

Magnus Perninge
Linnaeus University
59 PUBLICATIONS **533** CITATIONS

SEE PROFILE

Luigi Vanfretti
Rensselaer Polytechnic Institute
319 PUBLICATIONS **3,654** CITATIONS

SEE PROFILE

2013 IREP Symposium-Bulk Power System Dynamics and Control IX (IREP), August 25-30, 2013, Rethymnon, Greece

Approximating a Post-Contingency Stable Operation Region in Parameter Space through Time-Domain Simulation

Magnus Perninge, Jan Lavenius and Luigi Vanfretti
Royal Institute of Technology, Sweden

Abstract

Operating criteria for power systems, such as the $N-1$-criterion, are often based on evaluating whether the system is vulnerable to a specific set of contingencies. Therefore, a major part of power system security is concerned with establishing regions in parameter space where the system is vulnerable to specific contingencies. In this article we exploit the possibility of using Monte Carlo simulations to build an approximation of the region, in parameter space, where the power system will remain stable post a contingency.

Introduction

Due to the increased utilization of the power grid, that has followed upon the deregulation of electricity markets, security considerations now play a prominent role in power system operation [1]. There is a conflict of interest between the desire to transfer large amounts of electric energy through the power grid, and the requirement of secure operation where the risk of a disturbance is small. To satisfy both objectives to the largest possible extent an "adequate balance" between secure operation and transfer capacity is preferable.

Maintaining a secure operation of the power system is often considered equivalent to the system being able to withstand each and every contingency in a list of plausible contingencies. This requires knowledge of for what parameter levels the system is vulnerable to different contingencies. When the list of plausible contingencies is not too long and the system parameters can be forecasted with a good accuracy on-line, or close to on-line, time-domain simulations can be used the predict whether the system satisfies the security criteria. However, with the growing penetration of variable energy sources, that make the system parameters more difficult to predict, and the more complicated interconnections we see as a result of meeting the deregulation requirements, extensive on-line simulations will not be a valid tool.

One way to make detailed analysis tractable would be to perform dynamic simulations off-line to predict which contingencies that are important to investigate on-line. Another approach, which is the one we suggest in this paper is to make detailed simulations off-line and try to estimate a stability region which can then be used to steer the system on-line. The on-line steering can be performed by, for example, changing the production according to the solution of a security-constrained optimal power flow (SCOPF) (see [2] for the original work or [3,4] for more recent versions) or a stochastic optimal power flow (SOPF) [5, 6]. In [6] a SOPF formulation including the expected security cost from post-contingency corrective actions, such as fast generator rescheduling and load shedding, was given. The problem, which we will refer to as expected security cost stochastic optimal power flow (ESCSOPF), was formulated as:

Problem (ESCSOPF). Given $\alpha > 0$ and a list of n_c contingencies each with probability q_i, $i = 1, \ldots, n_c$, of occurrence; solve

$$\min_{u \in U} \quad \sum_{i=0}^{n_c} q_i E[C_i(u, Z)] \tag{1a}$$

$$\text{s.t.} \quad \sum_{i=0}^{n_c} q_i P[\min_{j \in \mathcal{J}_i^{TS}} d_{ij}(u, Z) < 0] \leq \alpha, \tag{1b}$$

where $q_0 = 1$, u is a vector of controllable parameters, Z is a random vector of uncertain parameters, $C_0(u)$ is the base case operating cost, $C_i(u, Z)$ is the security cost required to meet the mid-term stability requirements (see [7]) for contingency $i = 1, \ldots, n_c$, and $d_{ij}(u, Z)$ is the distance to the j^{th} ($j \in \mathcal{J}_i^{TS}$) second order approximation part of the boundary of the region where the system retains transient stability following contingency i.

This type of problem also appear under the name chance constrained optimal power flow CCOPF in the literature [8].
In order to solve this problem we need to first find a set $\{\Sigma_{ij}\}_{j \in \mathcal{J}_i^{\Pi}}$ of second order approximations that combined give a good picture of the boundary that separates the region where the system retains transient stability and the region where transient stability is lost post contingency i, for $i = 1, \ldots, n_c$. Our tool to obtain this will be repeated time-domain simulations in a Monte Carlo simulation followed by an identification of boundary points of the stable set and finally an approximation of the boundary using a number of second order polynomials.

Stability regions

For each contingency i, the dynamic stability region can be seen as the intersection of two subregions, one region D^i_{TS} where the system retains transient stability, and one region D^i_{MT} where mid-term stability [9] properties hold.

The mid-term stability region D^i_{MT}

Approximations of the boundary ∂D^i_{MT} of the domain D^i_{MT} have been proposed in the literature before [10–13]. In [11] the closest bifurcation point from a loading point is calculated and in [10], sensitivities of the loading margin (distance to the mid-term stability boundary) with respect to the system parameters are given. The use of the sensitivities can help the system operator take optimal actions to either steer the system away from instability or to make it stable again. In [12–15] second order approximations of the loadability surface are derived. In [16] a third order approximation is derived and a method for handling the intersections of saddle-node bifurcations and switching loadability limits is proposed.
Since much work has already been done on trying to approximate the mid-term stability boundary we will focus on approximating the boundary of the domain where the system remains transiently stable following a specific contingency.

The transient stability region D^i_{TS}

We assume that we want to investigate the post contingency stable region for contingency i. To do this we first generate a set $\Gamma = \{\lambda_1, \ldots, \lambda_M\}$ of points in a Monte Carlo simulation. Let D^i_{TS} be the set of all points in parameter space where the system retain transient stability after being subject to contingency i and let $\Gamma^i_{\text{TS}} = \{\lambda_{\text{TS}_i(1)}, \ldots, \lambda_{\text{TS}_i(n_{\text{TS}})}\} = \Gamma \cap D^i_{\text{TS}}$. The set Γ^i_{TS} is thus the collection of points of Γ for which a time domain simulation results in a transiently stable trajectory when applying contingency i. Note here that the post contingency trajectory should include the control actions that are normally used to save the system from loss of stability.
Assuming that the stable region is convex we could estimate D^i_{TS} with the convex hull, $\text{Conv}(\Gamma^i_{\text{TS}})$, of Γ^i_{TS}. Note that Γ^i_{TS} and thus also $\text{Conv}(\Gamma^i_{\text{TS}})$ is a random set. In Fig. 1 the result of disconnecting 60 MW of production in Generator 2 of the WSCC 9-bus system is shown for 1000 randomly selected values of the loads at nodes 5 and 8 while keeping a constant load at node 6.

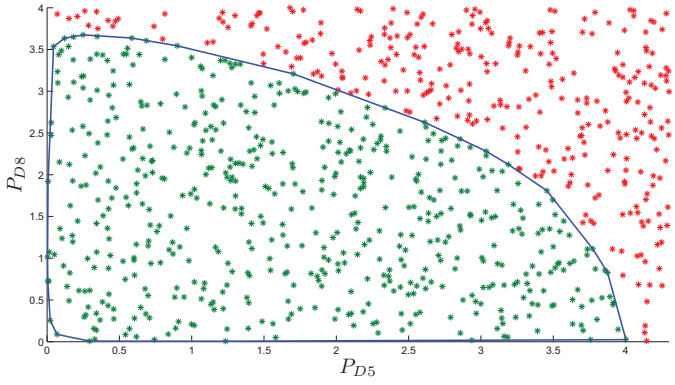

Figure 1: The stable domain of the WSCC 9 bus system for a 60 MW outage in Generator 2.

In the figure the green dots are cases for which the system retained transient stability (*i.e.* the set Γ^i_{TS} is given by the green dots) and the red dots correspond to cases where transient stability was lost. The area contained in the polygon is the convex hull $\text{Conv}(\Gamma^i_{\text{TS}})$ of the set of green points.
Unfortunately, we cannot assume that D^i_{TS} is convex and we might thus have that $\text{Conv}(\Gamma^i_{\text{TS}}) \not\subset D^i_{\text{TS}}$. To solve this we will use a triangular representation of the domain $\text{Conv}(\Gamma^i_{\text{TS}})$ and remove triangles if necessary to make it fit better with D^i_{TS}.
In a m-dimensional parameter space, assume that $m+1$ stable points $\{\lambda^{\text{TS}}_1, \ldots, \lambda^{\text{TS}}_{m+1}\}$ have been detected. If these points are not lying in a plane the convex hull of these points will form a m-simplex of positive volume. The m-simplex is bounded by $m+1$ hypersurfaces and any point λ within the simplex will satisfy

$$\begin{aligned} n_1^T(\lambda - p_1) &\leq 0 \\ &\vdots \\ n_{m+1}^T(\lambda - p_{m+1}) &\leq 0, \end{aligned}$$

where n_i and p_i are the normal to and a point of the i^{th} bounding hupersurface, respectively. When adding an additional point $\lambda^{\text{TS}}_{m+2}$, we can get two different situations. Either the new point is inside the m-simplex formed by the points $\{\lambda^{\text{TS}}_1, \ldots, \lambda^{\text{TS}}_{m+1}\}$ or the new point is outside the m-simplex.
The first situation is depicted in Fig. 2 for a 2-simplex (*i.e.* a triangle). As can be seen from the figure the convex hull does not

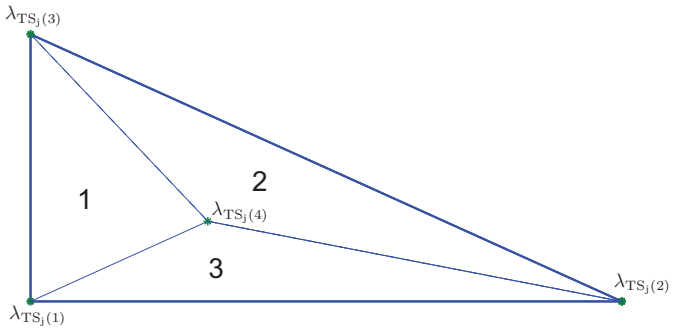

Figure 2: The approximation of the domain does not change when a new stable point is found inside the domain.

change when a new point inside is discovered. Rather the existing m-simplex is divided into a simplicial complex with $m+1$ smaller m-simplexes. These new m-simplexes are formed by taking the new point $\lambda^{\text{TS}}_{m+2}$ and making a m-simplex together with m of the points $\{\lambda^{\text{TS}}_1, \ldots, \lambda^{\text{TS}}_{m+1}\}$. As this can be done in $m+1$ different ways we get $m+1$ new (smaller) m-simplexes.
The other situation is when the new point lies outside of the initial m-simplex. This situation is depicted in Fig. 3 for the same 2-simplex that was used above. To form a simplicial complex with the $m+2$ points we add a number of new m-simplexes, one for each bounding surface which $\lambda^{\text{TS}}_{m+2}$ lies on the outside of. If $\lambda^{\text{TS}}_{m+2}$ is outside of bounding surface k, *i.e.*

$$n_k^T(\lambda^{\text{TS}}_{m+2} - p_k) > 0, \tag{2}$$

then a new simplex with vertices $\lambda^{\text{TS}}_{m+2}$ and the m points of $\{\lambda^{\text{TS}}_1, \ldots, \lambda^{\text{TS}}_{m+1}\}$ that lie in bounding surface i are used to form

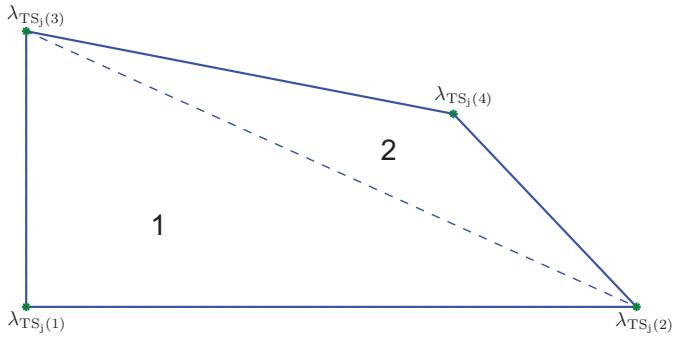

Figure 3: A new stable point is found outside of the first m-simplex.

a new m-simplex. In this way we get the convex hull of the set $\{\lambda_1^{\mathrm{TS}}, \ldots, \lambda_{m+2}^{\mathrm{TS}}\}$.

So far we have only searched for the convex hull of a set of stable points. However, as noted above D_{TS}^i is not necessarily convex. Fig. 4 shows the stable and unstable points for a short circuit at node 4 followed by a disconnection of the line between nodes 4 and 7 in the WSCC 9-bus system. In this figure the

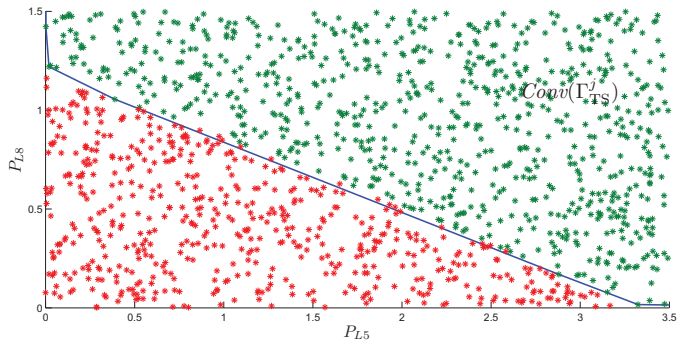

Figure 4: A sample of $\mathrm{Conv}(\Gamma_{\mathrm{TS}}^i)$ of the WSCC 9 bus system for a short circuit at bus 4 followed by a disconnection of the line between nodes 4 and 7. Green points are transiently stable and red are unstable.

higher loading boundary (which is not in the picture) is due to feasibility of the initial (pre-contingency) operating point and the lower level is due to transient instability. As we can see in the figure there are a number of parameter vectors leading to transient instability within the convex hull of the points where transient stability is retained.

In the view of this result we need to somehow update the approximation $\hat{D}_{\mathrm{TS}}^i$ of D_{TS}^i by removing m-simplexes so that no parameter vectors leading to transient instability are inside $\hat{D}_{\mathrm{TS}}^i$. Assume that after sampling k parameter vectors we have built the approximation $\hat{D}_{\mathrm{TS},k}^i$ and the $(k+1)^{th}$ sample lies within $\hat{D}_{\mathrm{TS},k}^i$ and the corresponding time domain simulation gives a transiently unstable path. Then we have to remove the smallest m-simplex that contains the unstable point.

In Fig. 5 we see what happens when a unstable point is encountered inside the convex hull in the example given in Fig. 2. In this case the approximation of the stable domain is reduced to

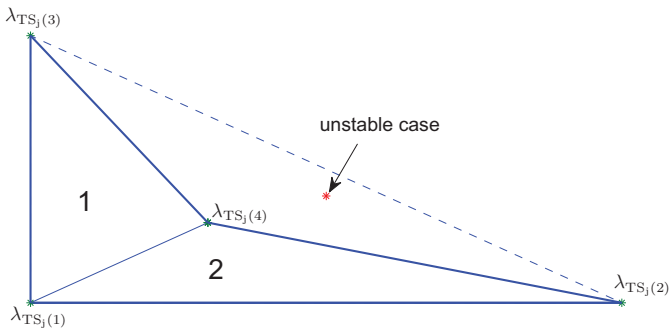

Figure 5: When an unstable point inside the approximation is found we remove the triangle that contains the point.

the two triangles labeled 1 and 2.

When all the samples are examined we should further investigate the boundary by verifying that the middle of each part of the boundary corresponds to a stable path in a time-domain simulation.

The resulting domain is given by the condition

$$(n_k^{\mathrm{CH}})^T(\lambda - p_k^{\mathrm{CH}}) \leq 0, \quad k = 1, \ldots, N_{\mathrm{CH}}, \quad (3)$$

$$\max_{j \in \{1,\ldots,m+1\}} (n_j^{R,k})^T(\lambda - p_j^{R,k}) \geq 0, \quad k = 1, \ldots, N_R, \quad (4)$$

where n_k^{CH} and p_k^{CH} are the outward normals of and points of the boundary of the convex hull, respectively, N_{CH} are the number of boundary parts of the convex hull, $n_k^{R,j}$ and $p_k^{R,j}$ are the outward normals and points of the $m+1$ bounding surfaces of the N_R removed m-simplexes.

Constraint (4) only holds true if the λ is inside the complex hull and constraint (3) is only true if λ is not lying within any of the removed m-simplexes.

Fig. 6 shows the updated approximation of the transient stability domain obtained by first removing all triangles that contained unstable points. Each part of the resulting boundary was then investigated by finding the middle of the boundary part and performing a time-domain simulation to validate that the boundary is part of the transient stability domain.

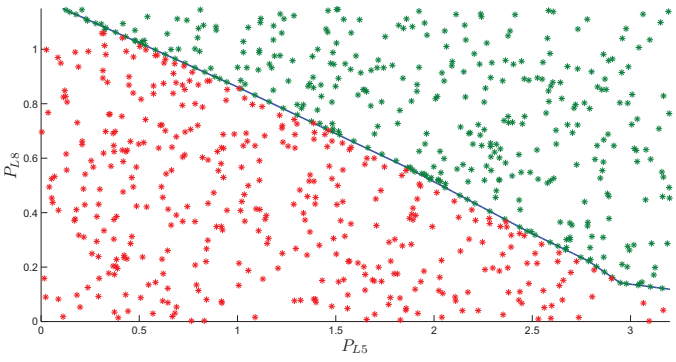

Figure 6: The domain $\hat{D}_{\mathrm{TS}}^i$ after removing the triangles that contain unstable points and checking the new boundary parts.

Second order approximations of the transient stability region

The procedure for determining approximations of the boundary ∂D^i_{TS} of the domain D^i_{TS} is as follows:

1. Perform a number of Monte Carlo simulations where the transient stability properties for a randomized parameter vector is investigated by time-domain simulation.

2. From the simulated points approximate the boundary of the stable domain.

3. Approximate this boundary using several second order polynomials in the system parameters.

The two first steps have already been explained above. What remains is to compute second order polynomials that together well approximate the boundary of the transient stability region.

To get a good approximation of the boundary, ∂D^i_{TS}, of the transient stability region several polynomials will often be needed. This is illustrated in Fig. 7.

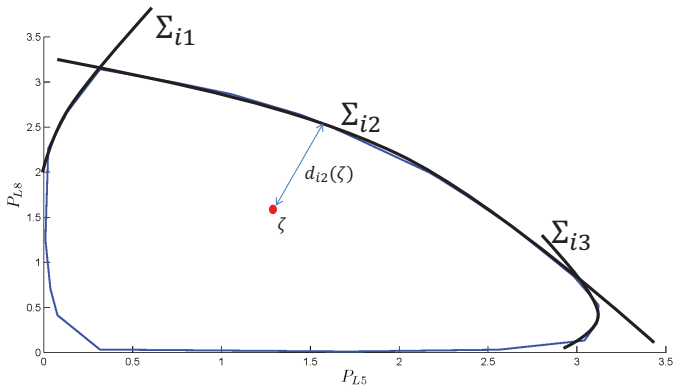

Figure 7: Polynomial approximations of ∂D^i_{TS}.

Since second order polynomial approximations were suggested earlier we need to decide vectors $\lambda_c^{ij} \in \mathbb{R}^m$ where the approximation has its basis, normals $n^{ij} \in \mathbb{R}^m$ and curvature tensors A^{ij} that are symmetric $m \times m$–matrixes such that $A^{ij} n^{ij} = 0$.
Assume the we have a prediction of future system parameters giving the probability density function $f_\Theta : \mathbb{R}^m \to \mathbb{R}_+$. To get a good accuracy of the approximation we choose λ_c^{i1} to be the boundary point that maximizes f_Θ, *i.e.* the most probable point of the boundary.
Now, since the boundary has not been calculated exactly, the boundary will probably not be smooth and local information at λ_c^{il} is often not that useful. Moreover, since λ_c^{il} was decided by maximizing a probability density function we may expect that λ_c^{il} is closer to the middle of the set than its neighbors on the boundary. Hence, when deciding $n^{ij} \in \mathbb{R}^m$ and A^{ij} we should discard all other boundary points lying on a distance $< \delta$ from λ_c^{il}. The constant $\delta > 0$ should be set depending on the accuracy (*e.g.* number of samples) of the initial boundary approximation.

Once the basis point λ_c^{il} and $\delta > 0$ have been decided we define

$$d_{il}(\lambda, n, A) = n^\top(\lambda_c^{il} - \lambda) + \frac{1}{2}\lambda^\top A\lambda. \tag{5}$$

This means that the point λ is on the inner side (in negative direction of the normal n) of the codimension-one surface defined by $\hat{\Sigma}_{ij}(n, A) := \{\lambda \in \mathbb{R}^n : d_{ij}(\lambda, n, A) = 0\}$ whenever $d_{ij}(\lambda, n, A) > 0$.
Let Λ_0^i be the set of all boundary points computed in step 2. The point λ_c^{il} of Λ_0^i that maximized $f_\Theta : \mathbb{R}^m \to \mathbb{R}_+$ was picked to be the basis for the first approximation. We now remove all the points on a distance $< \delta$ to λ_c^{il} and get the set $\Lambda_0^i(\lambda, \delta) = \Lambda_0^i \cap (B_\delta(\lambda))^c$, where $(B_\delta(\lambda))^c$ is the complement of the δ-ball centered at λ. We now want to choose n and A, so that $d_{ij}(\lambda, n, A) \geq 0, \forall \lambda \in \Lambda_0^i(\lambda, \delta)$. But still we want the fit to be as close as possible. We thus pose the following optimization problem:

$$\min_{n \in \mathbb{S}^{m-1}, A \in \text{Sym}_\text{m}} \quad \sum_{\lambda \in \Lambda_0^i} f_\Theta(\lambda) d_{il}(\lambda, n, A) \tag{6a}$$

$$\text{s.t.} \quad d_{il}(\lambda, n, A) \geq 0, \ \forall \lambda \in \Lambda_0^i(\lambda_c^{il}, \delta), \tag{6b}$$

$$An = 0. \tag{6c}$$

Note here that once we have picked a $n \in \mathbb{S}^{m-1} = \{x \in \mathbb{R}^m : \|x\| = 1\}$, (6) is a linear program. A heuristic way of solving (6) would be to set $\hat{n} = (\lambda_c^{il} - \zeta)/\|\lambda_c^{il} - \zeta\|$ and let C be a $m \times (m-1)$–matrix with basis vectors of $\perp \hat{n}$. The distance from λ to the surface $\{\lambda_c^{il} + Cx + \hat{n}(a^\top x + 1/2 x^\top b x) \in \mathbb{R}^m : x, a \in \mathbb{R}^{m-1}, \ b \in \text{Sym}_{\text{m}-1}\}$ in the direction $\hat{n}$ is then

$$\hat{d}_{il}(\lambda, a, b) = \hat{n}^\top(\lambda_c^{il} - \lambda) + a^\top C^\top(\lambda_c^{il} - \lambda) + \frac{1}{2}(\lambda_c^{il} - \lambda)^\top C b C^\top(\lambda_c^{il} - \lambda).$$

If we instead of (6) solve

$$\min_{a \in \mathbb{R}^{m-1}, b \in \text{Sym}_{\text{m}-1}} \quad \sum_{\lambda \in \Lambda_0^i} \hat{f}_\Theta(\lambda) d_{il}(\lambda, a, b) \tag{7a}$$

$$\text{s.t.} \quad \hat{d}_{il}(\lambda, a, b) \geq 0, \ \forall \lambda \in \Lambda_0^i(\lambda_c^{il}, \delta), \tag{7b}$$

and repeatedly set $\hat{n} \leftarrow (\hat{n} + Ca)/\|\hat{n} + Ca\|$ and solve (7) we get the solution to (6) as a fixed point to the algorithm with $n^* = \hat{n}$ and $A^* = C^\top b^* C$.
Once problem (6) has been solved we set $d_{il}(\lambda) = d_{il}(\lambda, n^*, A^*)$, where (n^*, A^*) is the pair giving the optimal solution to (6). It might be worthwhile here to note that (6) is not a convex problem so having a good initial guess is important.
Now, we pick a second number $\gamma > 0$ and set $\Lambda_1^i = \{\lambda \in \Lambda_0^i : d_{il}(\lambda) > \gamma\}$. The set Λ_1^i will thus consist of all points of the boundary for which the first approximation has an accuracy less than γ. To compute the second part of the polynomial approximation of the transient stability boundary we look for the most probable point of the remaining non-covered part of the boundary. Hence,

$$\lambda_c^{i2} = \underset{\lambda \in \Lambda_1^i}{\text{Argmax}} \ f_\Theta(\lambda). \tag{8}$$

Similar to computing the first approximation the second is found by solving

$$\min_{n\in\mathbb{S}^{m-1},A\in\mathrm{Sym_m}} \quad \sum_{\lambda\in\Lambda_1^i} f_\Theta(\lambda) d_{\not 2}(\lambda,n,A) \tag{9a}$$

$$\text{s.t.} \quad d_{\not 2}(\lambda,n,A)\geq 0,\ \forall\lambda\in\Lambda_0^i(\lambda_c^{\not 2},\delta), \tag{9b}$$

$$An=0. \tag{9c}$$

This process is then repeated until

$$\max_{\lambda\in\Lambda_j^i} f_\Theta(\lambda)$$

is sufficiently small.

Example

In the numerical example we continue working with the two contingencies on the WSCC 9 bus system (see Fig. 8) defined above. The first contingency is an outage leading to a loss of 60 MW

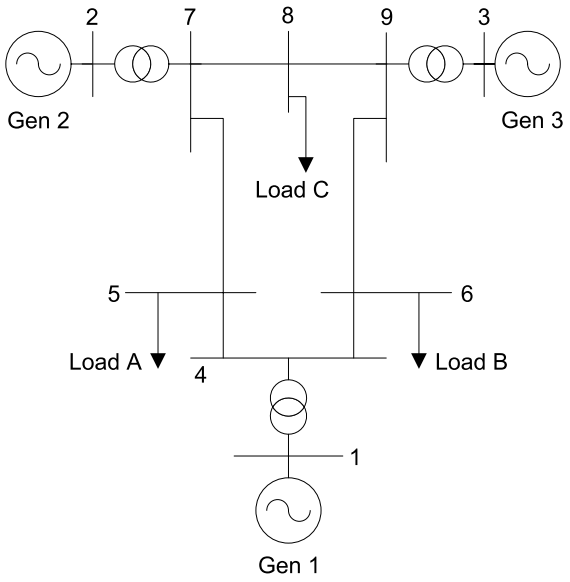

Figure 8: The WSCC 9 bus system.

production in the generator located at node 2. The second contingency is a 0.083 second short circuit at node 4 followed by a disconnection of the line connecting nodes 4 and 7.

We assume that the varying parameters are the active power loads at nodes 5 and 8, so that $\lambda = [P_{D5}\ P_{D8}]$. To create Γ we generate 1000 samples of a pair of independent random variables uniformly distributed on $[0, 5]$.

Assume that Θ is Gaussian with mean

$$m_\Theta = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

and variance

$$\mathrm{Var}\left[\Theta\right] = \begin{bmatrix} 0.75 & 0 \\ 0 & 1 \end{bmatrix}.$$

To build the boundary we will split all boundary segments until the length (Euclidean norm) of the longest segment is less than 0.1 p.u. For each of the new boundary points, time domain simulations have to be made to validate the stability. Once the stability of all boundary points is validated we start building the second order approximations. The result in the first contingency case with parameters $\delta = 0.3$ and $\gamma = 0.1$ is shown in Fig. 9. In

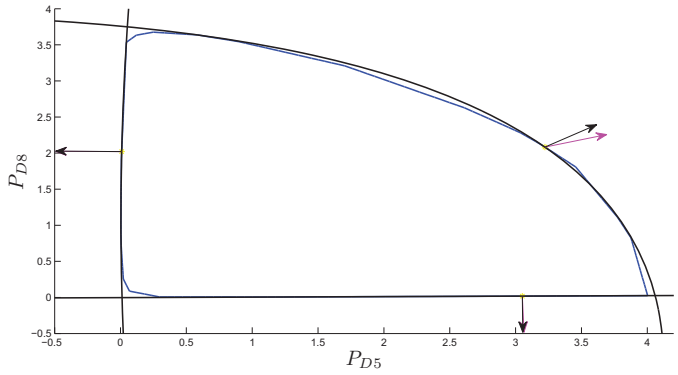

Figure 9: The approximation of the transient stability boundary for contingence $i = 1$ with $\delta = 0.3$ and $\gamma = 0.1$.

the figure the blue solid polygon is the transient stability boundary approximated from the points of the set Γ. In this case the transient stability domain seems to be convex so the convex hull of the points Γ_{TS}^i gives the initial approximation. From the set of boundary points we then need three second order approximations (black solid lines) to obtain the accuracy $\gamma = 0.1$. The magenta arrows show the initial guesses of the normal vectors and the black arrows are the normal vectors obtained by the iterative procedure described above. To get the initial guesses we used $\zeta = m_\Theta$.

To increase the accuracy of the approximations we set $\gamma = 0.05$ and $\delta = 0.2$. With these values we get the approximation shown in Fig. 10. For both figures we use a stopping criteria based on

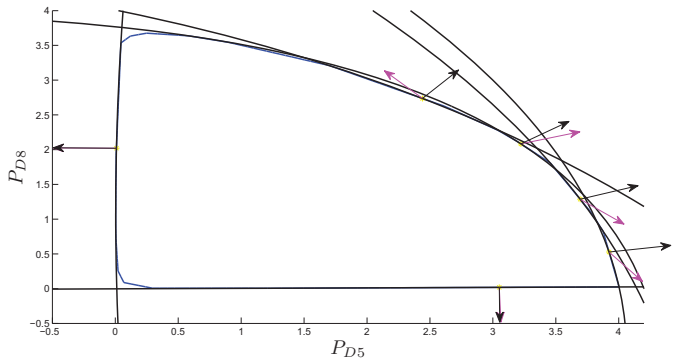

Figure 10: The approximation of the transient stability boundary for contingence $i = 1$ with $\delta = 0.2$ and $\gamma = 0.05$.

the likelihood of the remaining boundary points that prevents us from approximating the far off corners with an excessive degree of accuracy. With the increased requirement on the accuracy we see that we need a total of six second order approximations. As expected we thus see that increasing the accuracy requirements leads to the use of more approximations.

Discussion

In this paper we investigated how Monte Carlo simulations combined with time domain simulations can be used to build second order approximations of the transient stability boundary for a given list of contingencies. These approximations can then be combined with approximations of mid-term stability regions and other operational regions when solving an optimal power flow problem with chance constraints.

To compute the second order approximations the boundary points of the region, in parameter space, where the system remained stable after the contingency were identified. The second order approximations of this boundary were then obtained by solving a number of linear programming problems in an iterative approach.

Possible improvements of the method include:

Sequential scenario generation

If the stability of each scenario is checked directly after the scenario has been generated in the Monte Carlo simulation this information can be used to update the distribution from which we sample the future scenarios. One could, for example, use a set of indexes to evaluate "how stable" a stable point is or "how unstable" an unstable point is. This information can then be used to avoid drawing unnecessary samples and help us focus on the regions where the boundary between stable and unstable points is likely to appear.

Growing the region

When a first approximation of the transient stability region is obtained we could take small steps in the normal directions from the different affine parts of the boundary. This would give an efficient way of further expanding the boundary to get a maximal approximation of the stability region.

Non-heuristic guess of normal vector

The initial guess of the normal vector at the approximation points is very heuristic. One alternative could be to solve a linear problem where we maximize the distance of the closest boundary point to a hyperplane passing through the approximation point. The normal to this hyperplane can then be taken as a first approximation of the normal to the second order surface at the boundary point.

Higher order surfaces

Instead of increasing the number of approximating surfaces we may want to increase the order of the approximations. Note that this would preserve the linearity of the optimization problems but dramatically increase the number of variables when the parameter space has a high dimensionality.

References

[1] IEEE/CIGRE Joint Task Force on Stability Terms and Definitions. Definition and classification of power system stability. *Power Systems, IEEE Transactions on*, 19(3):1387–1401, 2004.

[2] O. Alsac and B. Stott. Optimal load flow with steady-state security. *Power Apparatus and Systems, IEEE Transactions on*, 93(3):745–751, 1974.

[3] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel. Contingency filtering techniques for preventive security-constrained optimal power flow. *Power Systems, IEEE Transactions on*, 22(4):1690 –1697, 2007.

[4] F. Capitanescu, T. Van Cutsem, and L. Wehenkel. Coupling optimization and dynamic simulation for preventive-corrective control of voltage instability. *IEEE Trans. Pwr. Syst.*, 24(2):796–805, 2009.

[5] I. Dobson, S. Greene, R. Rajaraman, C. L. DeMarco, F. L. Alvarado, M. Glavic, J. Zhang, and R. Zimmerman. Electric power transfer capability: concepts, applications, sensitivity and uncertainty. Technical report, Power Systems Engineering Research Center (PSERC), 2001.

[6] M. Perninge and C. Hamon. A stochastic optimal power flow problem with stability constraints; part II: The optimization problem. *IEEE Trans. Power Syst.*, 28(2):1849–1857, 2013.

[7] J. Condren, T. W. Gedra, and P. Damrongkulkamjorn. Optimal power flow with expected security costs. *IEEE Transactions on Power Systems*, 21(2):541–547, 2006.

[8] H. Zhang. Chance constrained programming for optimal power flow under uncertainty. *IEEE Trans. Power Syst.*, 26(4):2417–2424, 2011.

[9] T. Van Cutsem and C. D. Vournas. Voltage stability analysis in transient and mid-term time scales. *IEEE Transactions on Power Systems*, 11(1):146–154, 1996.

[10] S. Green, I. Dobson, and F. L. Alvarado. Sensitivity of transfer capability margins with a fast formula. *IEEE Transactions on Power Systems*, 17:34–40, 2002.

[11] F. Alvarado, I. Dobson, and Yi Hu. Computation of closest bifurcations in power systems. *IEEE Transactions on Power Systems*, 9(2):918–928, 1994.

[12] M. Perninge and L. Söder. On the validity of local approximations of the power system loadability surface. *IEEE Transaction on Power Systems*, 26(4):2143–2153, 2011.

[13] C. Hamon, M. Perninge, and L. Söder. A stochastic optimal power flow problem with stability constraints; part I: Approximating the stability boundary. *IEEE Trans. Power Syst.*, 28(2):1839–1848, 2013.

[14] M. Perninge and L. Söder. Risk estimation of the distance to voltage instability using a second order approximation of the saddle-node bifurcation surface. *Electric Power System Research*, 81(2):625–635, 2011.

[15] M. Perninge and L. Söder. Geometric properties of the loadability surface at SNB-SLL intersections and tangential intersection points. In *16th International Conference on Intelligent System Application to Power Systems (ISAP)*, September 2011.

[16] M. Perninge. Approximating the loadability surface in the presence of SNB-SLL corner points. *Elect. Power Syst. Res.*, 96:64–74, 2013.