

# Phase Boundary Computation for Fault Induced Delayed Voltage Recovery

Michael W. Fisher

Ian A. Hiskens

**Abstract**—Distribution networks that supply large numbers of induction motors are vulnerable to Fault Induced Delayed Voltage Recovery. This phenomenon is usually triggered by a transmission fault but results in a delayed recovery of voltages in the distribution feeder, usually taking several seconds for a return to pre-fault conditions, if at all. The general mechanism underlying this delayed recovery arises from the coupled nonlinear dynamics of induction motors stalling. It is important to establish the phase boundary that separates parameters that lead to stalled versus unstalled motor states. This paper develops a novel algorithm, based on shooting methods and Euler homotopy continuation, for obtaining the phase boundaries. It forces a trajectory to spend a fixed amount of time near an unstable equilibrium, and then increases that time until the trajectory approaches the unstable equilibrium point arbitrarily closely. This technique does not require prior knowledge of the unstable equilibrium point. Numerically computed phase boundaries, in terms of induction motor moments of inertia, fault clearing times, and nonhomogeneous networks are presented. The techniques are formulated in generality, and could be applied to compute phase boundaries for a large class of dynamical systems.

## I. INTRODUCTION

Due in part to increased penetration of air conditioners, with their associated induction motors, a serious and growing concern in distribution systems is Fault Induced Delayed Voltage Recovery (FIDVR) [1]. A FIDVR event is typically initiated by a transmission-level fault near a distribution substation, and results in a momentary voltage reduction across the distribution network. The fault in the transmission system is usually cleared quickly, and the voltage at the distribution substation returns to normal, but voltage across the distribution network may be very slow to recover. This occurs because the momentary reduction in voltage leads to a reduction in the electrical torque of induction motors, with no corresponding reduction in mechanical torque, causing them to reduce their rotational speed or even to stall. Motors that are stalling or that are reaccelerating draw a large amount of current, which leads to local voltage reduction. Sometimes several motors stall as a result of the fault, and the system temporarily settles to a new steady state with voltages dangerously low. This is referred to as FIDVR. If these low voltages are sustained, protection systems may trip [1], leading to further stress on the system and possibly even voltage collapse.

M. Fisher and I. Hiskens are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA, [fishermw@umich.edu](mailto:fishermw@umich.edu), [hiskens@umich.edu](mailto:hiskens@umich.edu)

The authors acknowledge the support of the Los Alamos National Laboratory Grid Science Program subcontract 270958 and the National Science Foundation grant ECCS-1307754.

Despite the concerns raised by FIDVR, current modeling is unable to accurately capture the behavior or predict the sensitivity of FIDVR events to parameter variations. Of particular interest is identifying the boundary between sets of parameters that result in stalled versus unstalled conditions, referred to as the *phase boundary*. When two parameters are free to vary, this boundary is described by a curve. It is very useful to know this boundary so that FIDVR events can be better anticipated. Currently this is achieved through trial and error by running a large number of simulations.

Physical mechanisms underlying the cascading stalling of induction motors were discussed in [5], with an emphasis on the coupling of the nonlinear dynamics between a large number of induction motors. However, [5] assumes a continuous distribution of induction motors along a feeder, and employs brute force techniques to locate the phase boundary.

The approach adopted in this paper is to find an initial point on the phase boundary using a shooting method and then continue the point into a curve using an Euler homotopy continuation method. The key idea behind the shooting method is to force a trajectory to spend a prescribed amount of time near an unstable equilibrium point (UEP), and then to increase that time until the trajectory approaches the UEP arbitrarily closely. Since the trajectory will converge to the UEP as the specified time approaches infinity, in practice this algorithm forces the trajectory as close as desired to the UEP. Due to the generality of this formulation, these techniques can be readily applied to compute phase boundaries for a wide range of dynamical systems.

The remainder of the paper is organized as follows. Section II presents an overview of induction motor dynamics and trajectory sensitivities, Section III describes the algorithms for computing phase boundaries for unknown UEPs, Section IV provides an overview of the numerical experiments and Section V discusses the results. Conclusions and directions for future work are presented in Section VI.

## II. BACKGROUND

### A. Dynamics of an induction motor

The dynamic behavior of an induction motor arises principally from the swing equation,

$$2H\dot{\omega}_r = T_e - T_m$$

where  $H$  denotes the moment of inertia,  $\omega_r$  is the angular frequency of rotation of the rotor,  $T_e$  is the electric torque and  $T_m$  is the mechanical torque applied to the rotor. A common model for the mechanical torque in an induction motor [2]

is,

$$T_m = T_o \left( \frac{\omega_r}{\omega_s} \right)^m \quad (1)$$

where  $\omega_s$  is the nominal system angular frequency,  $T_o$  is the nominal torque and  $m \in [0, 1]$ . The electrical torque is given by

$$T_e = \frac{P_e}{\omega_s}$$

where  $P_e$  is the electrical power supplied to the rotor.

When analyzing induction motor dynamics, it is useful to make a coordinate transformation to a frame rotating at the nominal frequency  $\omega_s$ . This frame is called the  $d-q$  frame, and the voltages and currents in this frame are given by  $V_d + jV_q$  and  $I_d + jI_q$ . The electrical power in this frame is given by  $P_e = V_d I_d + V_q I_q$ . Converting to per unit in terms of the nominal frequency  $\omega_s$ , the electric torque and power are the same, which results in

$$2H \frac{\dot{\omega}_r}{\omega_s} = \left( V_d I_d + V_q I_q - T_o \left( \frac{\omega_r}{\omega_s} \right)^m \right). \quad (2)$$

In addition to the swing equation, standard relations among currents and voltages apply, adjusted for the  $d-q$  frame. These induce dynamics in the currents and voltages as a result of the varying motor frequency and of the rotating  $d-q$  reference frame. A full description is available in [2]. The examples presented later use this more complete third-order dynamic model.

The slip of a motor is defined as,

$$s = \frac{\omega_s - \omega_r}{\omega_s}. \quad (3)$$

The motor stalls when  $\omega_r$  reaches 0, or in other words when  $s = 1$ . It is not physically possible for the motor to recover from a stalled state.

Figure 1 shows the terminal voltage  $V_t = |V_d + jV_q|$  of a single induction motor for a fault occurring near the motor at 0.5s. The fault clearing time, i.e. the time between the occurrence and removal of the fault, is varied. As the fault clearing time increases, the system becomes more and more stressed and eventually stalls for a clearing time of 0.45s. Notice that as the clearing time increases the voltage remains low for a longer period of time before returning to its original value. The system is approaching the boundary of the region of attraction ever more closely as the clearing time increases. This trait motivates the phase boundary algorithm presented later.

### B. Trajectory sensitivities

Trajectory sensitivities describe the change in a trajectory resulting from perturbations in initial conditions<sup>1</sup>. The dynamic behaviour of a nonlinear differential-algebraic system,

$$\dot{x} = f(x, y), \quad x(0) = x_0, \quad (4)$$

$$0 = g(x, y), \quad (5)$$

<sup>1</sup>Sensitivity to perturbations in parameters  $\theta$  can also be captured by modeling parameters as “states” through the introduction of trivial differential equations  $\dot{\theta} = 0$ ,  $\theta(0) = \theta_0$ .

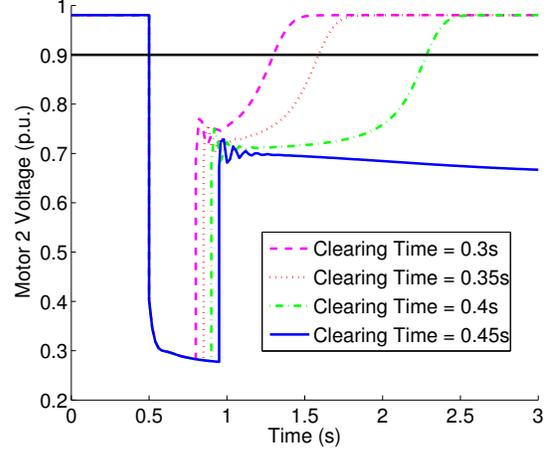


Fig. 1. Voltage behavior for a range of fault clearing times.

with  $x \in \mathbb{R}^n$  and  $y \in \mathbb{R}^m$ , can be expressed analytically by the flow,

$$x(t) = \phi(t, x_0) \quad (6)$$

together with (5). A Taylor series expansion of (6) gives,

$$\begin{aligned} \Delta x(t) &= \phi(t, x_0 + \Delta x_0) - \phi(t, x_0) \\ &\approx \frac{\partial \phi(t, x_0)}{\partial x_0} \Delta x_0 \end{aligned}$$

where  $\Phi(t, x_0) \triangleq \frac{\partial \phi(t, x_0)}{\partial x_0}$  is referred to as the *sensitivity transition matrix* or simply *trajectory sensitivities*. These concepts are extended to hybrid dynamical systems in [6] where the model allows for arbitrarily complicated switching conditions and incorporates state reset (jump) actions. Furthermore, trajectory sensitivities can be computed efficiently if an implicit numerical integration technique, such as trapezoidal integration, is used to establish the nominal trajectory. Full details are provided in [6].

Trajectory sensitivities will be used in forming the Jacobians in the shooting methods developed in Section III.

## III. ALGORITHMS

In order to identify a set of parameters that lies on the phase boundary, a trajectory is forced to remain near an (unknown) UEP for a prescribed length of time. This time is then increased, forcing the trajectory to approach the UEP, and hence lie arbitrarily close to the boundary of the region of attraction. Once a single point on the phase boundary has been identified, a continuation method based on a Euler homotopy can be used to numerically construct a 1-manifold or curve that describes the phase boundary in the plane of two free parameters.

### A. Shooting method for locating a target hypersurface

Prior work [7], reviewed below, has been successful in driving a system of DAEs to a target hypersurface at a fixed time. The novel contribution developed here is to increase this time towards infinity in order to force a trajectory to approach the boundary of the region of attraction.

For any fixed time  $\alpha$ , the system can be driven to a target hypersurface  $b(x, y) = 0$  by the following shooting method. Let  $\theta$  represent a free parameter of interest. Then the initial conditions are given by  $x_0 = x_0(\theta)$  and the flow by  $\phi = \phi(t, x_0(\theta))$ . We wish to find a root of the system of equations,

$$F_{s1}(x_a, \theta) = \phi(\alpha, x_0(\theta)) - x_a = 0 \quad (7)$$

$$F_{s2}(x_a, y_a) = g(x_a, y_a) = 0 \quad (8)$$

$$F_{s3}(x_a, y_a) = b(x_a, y_a) = 0 \quad (9)$$

where  $(x_a, y_a)$  is the desired intersection point with the target hypersurface. The variables are  $z = [x_a^T \ y_a^T \ \theta]^T$ . This denotes a system of  $n + m + 1$  equations in the same number of unknowns, and therefore the solution will consist of isolated point.

The system  $F_s(z) = 0$  given by (7)-(9) can be solved efficiently using Newton-Raphson with the following update

$$z^{n+1} = z^n - (DF_s(z^n))^{-1} F_s(z^n) \quad (10)$$

where the Jacobian is given by,

$$DF_s = \begin{bmatrix} -I & 0 & \Phi \frac{dx_0}{d\theta} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & 0 \\ \frac{\partial b}{\partial x} & \frac{\partial b}{\partial y} & 0 \end{bmatrix}. \quad (11)$$

Both the rate of convergence and the solution obtained are sensitive to the choice of initial conditions  $x_0(\theta)$ . However, the extensions to this algorithm that follow make it possible to reach the phase boundary even from an initial condition far from the boundary, so long as the initial condition results in a stable, unstalled steady state. For this stage of the algorithm, it is simplest to make an arbitrary guess for an initial condition, check the time at which it encounters the target hypersurface, and then choose a nearby target time with which to initialize the algorithm. The reasons for this choice will become clear in the following sections.

### B. Continuation method 1: finding a point on the phase boundary

The next stage of the algorithm is to increase the time at which the target hypersurface is reached until the system trajectory is arbitrarily close to instability. Let  $\alpha$  denote the target time for encountering the hypersurface  $b(x, y) = 0$ .

A homotopy continuation method is used to extend the target time in order to find a point on the phase boundary. To generate a homotopy path, it is necessary to free an additional parameter, which in this case will be  $\alpha$ . So,  $z = [x_a^T \ y_a^T \ \theta \ \alpha]^T$  giving a system of  $n + m + 1$  equations in  $n + m + 2$  unknowns, which defines a curve.

The continuation method proceeds by alternating between predictor and corrector steps. For a given iteration, suppose the initial starting point is  $z_0$ . The predictor step estimates a new point  $z_1 = z_0 + \kappa\eta$  where  $\eta$  is the unit tangent vector to the curve at  $z_0$  and  $\kappa$  is a scalar. For this part of the algorithm,  $\kappa$  is obtained using a backtracking line search which checks whether the simulation with the proposed new  $\theta$  leads to a stable, unstalled steady state. Then, the corrector

step projects the prediction  $z_1$  onto the true curve to obtain the next point. In this stage of the algorithm, the projection is done by fixing  $\alpha$  to its predicted value and using the shooting method described above to vary  $\theta$  until an intersection with the desired curve is obtained.

The algorithm terminates when the change in  $\theta$  between successive iterations is sufficiently small. When the algorithm terminates, a point on the desired phase boundary has been found.

### C. Continuation method 2: computing successive points on the phase boundary

Once a point on the phase boundary has been located, it can be extended to a curve using an Euler homotopy continuation method, similar to that described in [8] but with some modifications. Fix  $\alpha$  to its value at the end of the last iteration, and free an additional parameter  $\mu$ . Then  $z = [x_a^T \ y_a^T \ \theta \ \mu]^T$  so there are  $n + m + 2$  unknowns in  $n + m + 1$  equations, which will generate a curve. The predictor step is performed as in the homotopy continuation method above, using the unit tangent vector  $\eta$ , but the corrector step consists of two stages.

The inner stage involves fixing  $\alpha$  and solving for the intersection of the continuation curve with a hyperplane that passes through the prediction point and which, when restricted to the  $\theta - \mu$  plane, is orthogonal to  $\eta$ .<sup>2</sup> The need for restriction to the  $\theta - \mu$  plane arises in the outer stage of the correction step, and will be described shortly. From a prediction point  $z_1$ , performing the inner stage is achieved by solving,

$$F_{i1}(x_a, \theta) = \phi(\alpha, x_0(\theta)) - x_a = 0 \quad (12)$$

$$F_{i2}(x_a, y_a) = g(x_a, y_a) = 0 \quad (13)$$

$$F_{i3}(x_a, y_a) = b(x_a, y_a) = 0 \quad (14)$$

$$F_{i4}(z) = (z - z_1)_{[k:k+1]}^T \eta_{[k:k+1]} = 0 \quad (15)$$

where  $k = n + m + 1$  and the last equation expresses the desired orthogonality between a hyperplane containing the prediction point and  $\eta$  when restricted to the  $\theta - \mu$  plane. This can be solved efficiently using the Newton-Raphson update (10) to find a root of  $F_i$  with

$$DF_i = \begin{bmatrix} -I & 0 & \Phi \frac{dx_0}{d\theta} & \Phi \frac{dx_0}{d\mu} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & 0 & 0 \\ \frac{\partial b}{\partial x} & \frac{\partial b}{\partial y} & 0 & 0 \\ 0 & 0 & \eta_k & \eta_{k+1} \end{bmatrix}. \quad (16)$$

If the correction step involved only the inner stage, it would trace out a curve of constant  $\alpha$  which intersected the target hypersurface  $b(x, y) = 0$  always at the target time  $\alpha$ . However, in practice different target times are required for different sets of parameters in order to remain close to the phase boundary. In other words, as the parameter values  $\theta$  and  $\mu$  change, it may be necessary to adjust  $\alpha$  in order to remain close to instability. This was observed in practice

<sup>2</sup>This is in contrast to the correction step in [8] where orthogonality to the full vector  $\eta$  is maintained.

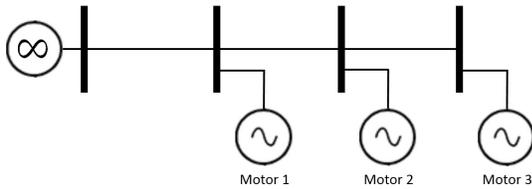


Fig. 2. Network used to simulate cascaded stalling. Three induction motors are arranged linearly with an infinite bus at the head.

for induction motor dynamics, as will be illustrated in later results, and motivates the need for the outer stage.

The outer stage involves multiplying  $\alpha$  by a factor greater than one until successive values of  $\theta$  and  $\mu$  differ from their values at the previous iteration by less than some specified tolerance. One interpretation of this stage is to consider a family of curves in  $\theta$  and  $\mu$  that is parameterized by  $\alpha$ . As  $\alpha$  approaches infinity, this family of curves converges to the true one-dimensional phase boundary. Typically, the outer stage is initialized with the value of  $\alpha$  obtained at convergence of the previous predictor-corrector step.

#### IV. NUMERICAL EXPERIMENTS

A simple radial test network was constructed, as depicted in Fig. 2. It consists of three identical induction motors connected to three buses arranged radially with an infinite bus at the head of the feeder. After initializing to steady state, a fault was applied at the bus closest to the infinite bus at 0.5s, and the ensuing dynamics were simulated. The voltage at the infinite bus was set to 0.98pu. The lines were all given resistances, reactances, and shunt admittances of 0.0069pu, 0.1090pu, and 1.106pu, respectively. The synchronous frequency was set at 60Hz. The rotor resistances were 0.0226pu, stator resistances were 0.0517pu, rotor reactances were 0.0800pu, stator reactances were 0.1058pu, motor shunt admittances were 3.6186pu, nominal torques were 0.822pu, and the exponents  $m$  were set to 0. The fault impedance was 0.05pu. These conditions represented a highly stressed situation for the motors.

The main parameters of interest in this study were motor moments of inertia and fault clearing time. Initial simulations were conducted on a network similar to that shown in Fig. 2 but with only two motors. This allowed trajectories to be depicted in the phase plane, with plots showing the slip of Motor 1 versus that of Motor 2. These simulations allowed exploration of the claims presented earlier regarding the necessity for the outer stage correction to  $\alpha$ . The phase plane plots also show the relevant UEP.

The target hypersurface was  $b(x, y) = V - 0.9$  where  $V$  represents the voltage of the motor furthest from the infinite bus in all cases. Phase boundaries were computed for two sets of free parameters, fault clearing time versus homogeneous moments of inertia, and the nonhomogeneous moment of inertia for Motor 1 versus that for Motor 2. Subsequently, similar boundaries were computed for the three motor system.

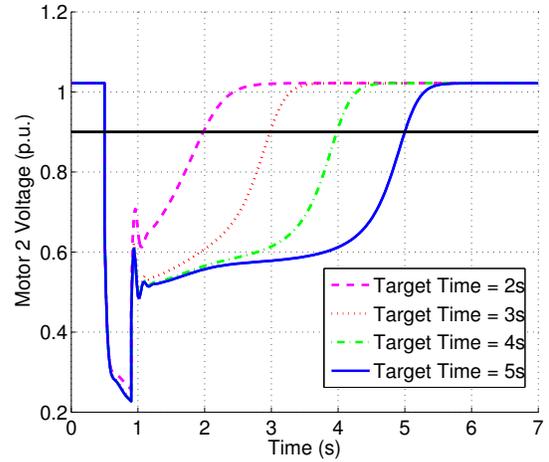


Fig. 3. Motor 2 voltage as a function of time for varying target times  $\alpha$ . The black line indicates the target hypersurface of 0.9V.

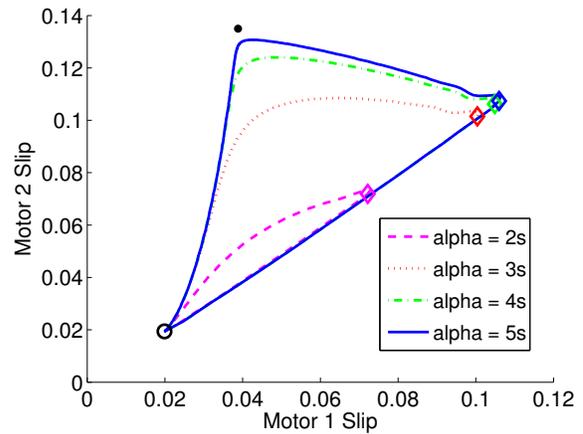


Fig. 4. Motors 1 and 2 state-space trajectories for increasing values of target time  $\alpha$ , homogeneous moments of inertia, and a fixed clearing time of 0.4s. The open circle indicates the initial steady state, the diamonds encircle the fault clearing points, and the closed circle shows the unstable equilibrium point. The trajectory proceeds counterclockwise.

#### V. RESULTS

##### A. Two induction motor network

The algorithms discussed above were designed under the assumption that increasing the target time  $\alpha$  would lead the trajectories to approach the UEP arbitrarily closely. Figure 3 shows Motor 2 voltage as a function of time for varying target times  $\alpha$ . Note that each trajectory encounters the target hypersurface of 0.9V exactly at its specified  $\alpha$ , as desired. Figure 4 shows the state-space trajectories of Motor 1 and Motor 2 slips for increasing values of target time  $\alpha$  with homogeneous moments of inertia and a fixed clearing time of 0.4s. The relevant UEP was computed and is shown as a black circle. As  $\alpha$  increases, the trajectories rapidly distort towards the UEP, justifying the assumption.

The continuation method 1 described above for computing an initial point on the phase boundary requires that the parameter varies monotonically as  $\alpha$  increases in order to

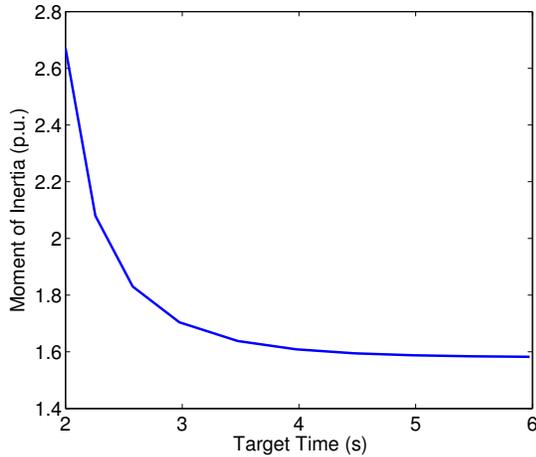


Fig. 5. Convergence of homogeneous moments of inertia to their critical value for increasing target time  $\alpha$ , at a fixed clearing time of 0.4s.

avoid becoming trapped in a local minimum. Figure 5 depicts the target time  $\alpha$  and the homogeneous moments of inertia as  $\alpha$  is increased for a fixed clearing time of 0.4s. The moment of inertia decreases monotonically with increasing  $\alpha$ , as expected, and converges approximately exponentially to its critical value.

As discussed earlier, the continuation method 2 requires an outer stage in order to ensure convergence to the true phase boundary. Figure 6 depicts Motor 2 voltage as a function of time for varying clearing times and corresponding moments of inertia. Note that each trajectory encounters the target hypersurface of 0.9V exactly at its specified time  $\alpha = 5s$ , as desired. The varying clearing times are evident near the start of the trajectories, and the different moments of inertia manifest in different curvatures within each trajectory. Figure 7 shows the state-space trajectories of Motor 1 and Motor 2 slips that correspond to the four cases depicted in Figure 6. As the clearing time and homogeneous moments of inertia increase, for a fixed target time  $\alpha$ , the trajectories lie further and further from the UEP.

As another illustration of this phenomenon, Figure 8 shows the state-space trajectories of Motor 1 and Motor 2 slips for increasing values of Motor 1 moment of inertia, and the corresponding decreasing values of Motor 2 moment of inertia, at the fixed target time  $\alpha = 5s$ . As the Motor 1 moment of inertia increases, for a fixed target time  $\alpha$ , the trajectories lie closer and closer to the UEP. However, note that the change in distance from the UEP is much smaller as a function of Motor 1 moment of inertia here than it was as a function of clearing time above. Accordingly, the changes in  $\alpha$  required to remain on the phase boundary are smaller as well.

The overall algorithm produces a one-dimensional phase boundary in the plane of two free parameters. This boundary separates stable sets of parameters from those that cause instability. Figure 9 depicts this boundary for the two motor network, where the free parameters are clearing time and

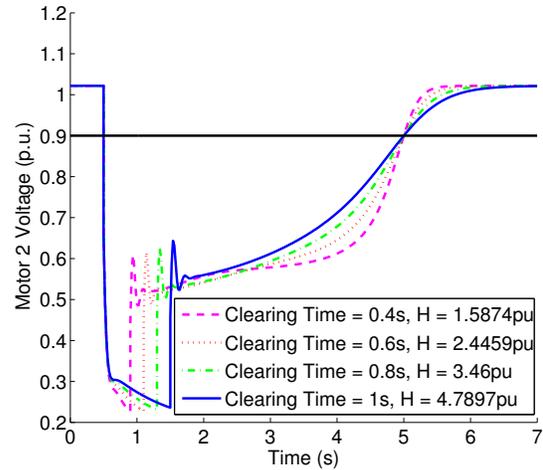


Fig. 6. Motor 2 voltage as a function of time for varying fault clearing times and corresponding moments of inertia  $H$ , at a fixed target time  $\alpha = 5s$ . The black line indicates the target hypersurface of 0.9V.

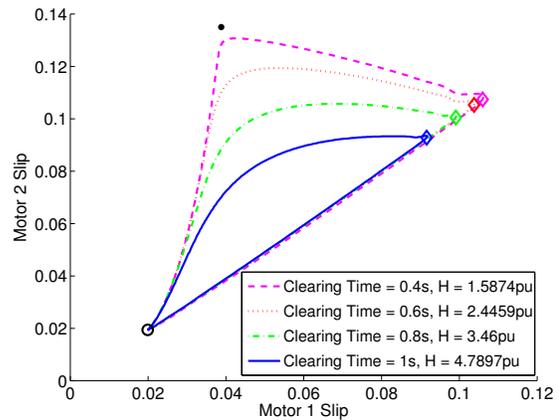


Fig. 7. Motors 1 and 2 state-space trajectories for increasing values of clearing time and homogeneous moments of inertia  $H$ , at a fixed target time  $\alpha = 5s$ . The open circle indicates the initial steady state, the diamonds encircle the fault clearing points, and the closed circle is the relevant UEP. The trajectory proceeds counterclockwise.

homogeneous moments of inertia. All points above this curve give stable behavior while those below result in instability. Note that the curve is almost linear. This is in agreement with prior work [5] based on a less realistic continuum approximation and found by trial and error. Figure 10 shows the target time  $\alpha$  required to remain on the phase boundary as a function of moment of inertia. It is necessary to increase  $\alpha$  substantially in order to remain on the boundary, as expected from the trajectory plots in Figure 7. Figure 11 shows the phase boundary corresponding to the free parameters Motor 1 and Motor 2 moments of inertia, at a fixed clearing time of 0.4s. All points above this curve give stable behavior while those below result in instability. This curve is more parabolic than the previous case. Only slight increases in  $\alpha$  were needed in this case to remain on the boundary. This is consistent with Figure 8 and contrasts with the above case

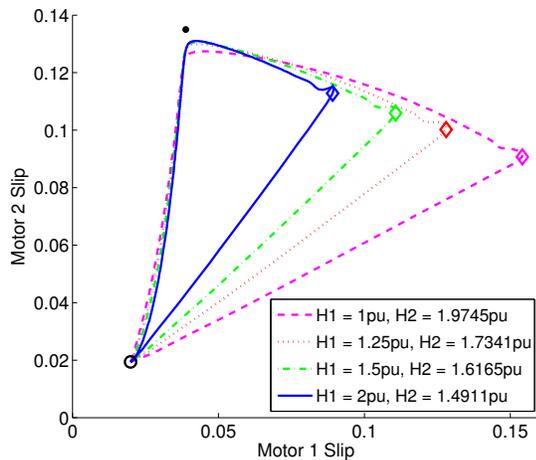


Fig. 8. Motors 1 and 2 state-space trajectories for increasing values of Motor 1 moment of inertia  $H_1$  and corresponding decreasing values of Motor 2 moment of inertia  $H_2$ , for fixed target time  $\alpha = 5$ s and fixed clearing time of 0.4s. The open circle indicates the initial steady state, the diamonds encircle the fault clearing points, and the closed circle is the relevant UEP. The trajectory proceeds counterclockwise.

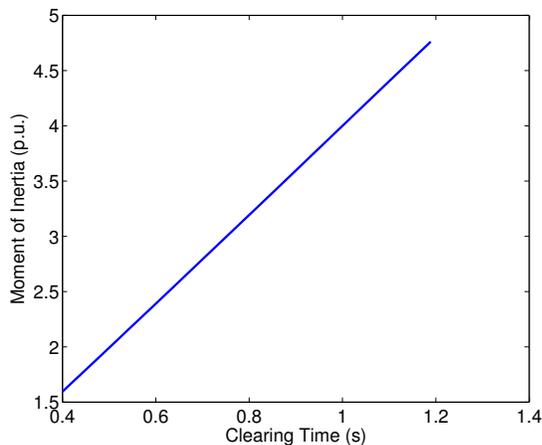


Fig. 9. Phase boundary for two motor network for varying clearing time and homogeneous moment of inertia.

of homogeneous moments of inertia and clearing time.

### B. Three induction motor network

The three motor network is already too large to be easily visualized in phase space. However, applying the same techniques as for the two motor setting results in similar phase boundaries and trends in target time  $\alpha$ . Figure 12 shows the phase boundary for the three motor network with the free parameters being the clearing time and homogeneous moments of inertia. As in the two motor case, it is almost linear. Figure 13 depicts the target time  $\alpha$  required to remain on the boundary as a function of homogeneous moment of inertia. As in the two motor case,  $\alpha$  increases substantially with the homogeneous moments of inertia. Figure 14 shows the phase boundary for free parameters Motor 2 and Motor 3 moments of inertia, at a clearing time of 0.25s and with

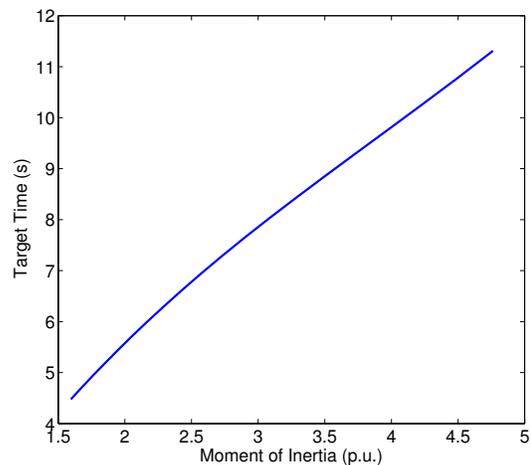


Fig. 10. Target time  $\alpha$  required to approach the UEP for two motor network as a function of homogeneous moment of inertia.

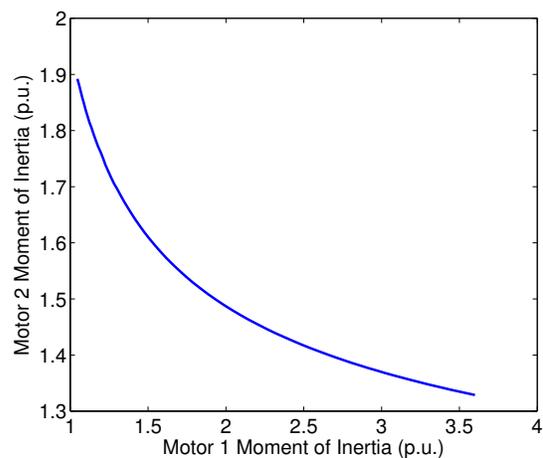


Fig. 11. Phase boundary of two motor network for varying nonhomogeneous Motor 1 and Motor 2 moments of inertia, at a fixed clearing time of 0.4s.

Motor 1 moment of inertia fixed at 2.8757pu. The curve is parabolic, as in the two motor case. The variations in  $\alpha$  required to remain on this boundary were small, as in the two motor case.

## VI. CONCLUSIONS AND FUTURE DIRECTIONS

A novel algorithm for numerical computation of one-dimensional phase boundaries has been presented. The algorithm was developed in generality and then applied to a test case motivated by the role of induction motors in fault induced delayed voltage recovery (FIDVR) events. A two-motor network was initially used to illustrate the different stages of the algorithm. The algorithm was then applied to a larger network where visualization was not straightforward. Phase boundaries that separated stable/unstable behavior were computed for pairs of free parameters, viz., fault clearing time and homogeneous moments of inertia, as well as pairs of nonhomogeneous moments of inertia. An almost

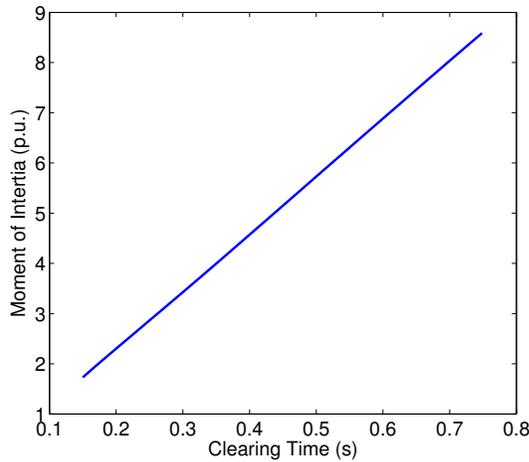


Fig. 12. Phase boundary for the three motor network for varying clearing time and homogeneous moment of inertia.

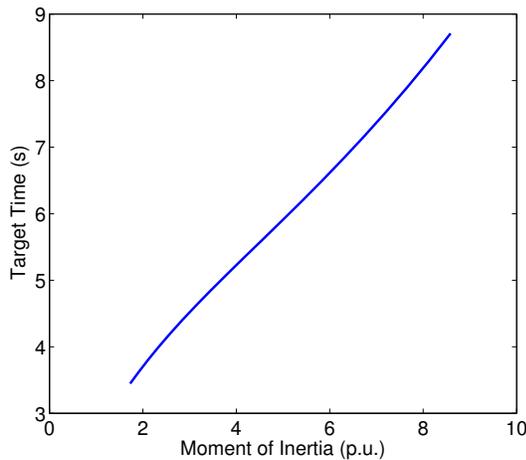


Fig. 13. Target time  $\alpha$  required to approach the UEP for the three motor network as a function of homogeneous moment of inertia.

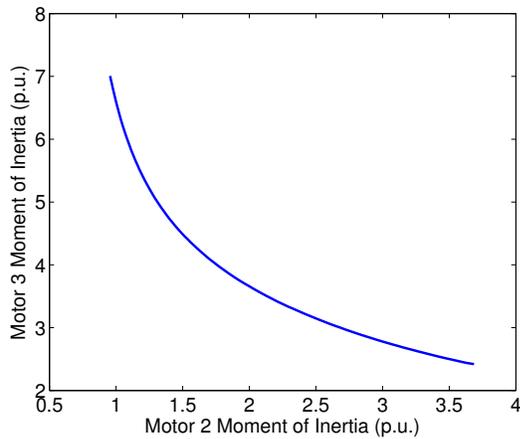


Fig. 14. Phase boundary of the three motor network for varying non-homogeneous Motor 2 and Motor 3 moments of inertia at fixed clearing time of 0.25s and Motor 1 moment of inertia of 2.8757p.u.

linear curve was found for the former while the latter resulted in a hyperbolic curve. This trend was observed for both the two- and three-motor networks.

There are several potential future directions. The phase boundary computation algorithm will be tested on larger networks with more complex topologies. The current requirement of varying the target time to remain on the boundary will be explored further. Another possibility is to compute the relevant UEP and force the trajectory to remain within a ball around that UEP for a fixed length of time. It is anticipated that this technique will not require as much adjustment of the target time as is the case for the current algorithm, although it will require prior knowledge of the UEP.

FIDVR is a dangerous phenomenon for distribution systems, and is of serious concern to power system operators. A better understanding of the parametric influences on stability for FIDVR events will enable better planning and more robust operation, leading to safer and more reliable distribution grids.

## REFERENCES

- [1] NERC Technical Reference Group. Fault-induced delayed voltage recovery (FIDVR). Technical report, NERC, 2009.
- [2] P. Kundur. Power System Stability and Control. New York: McGraw-Hill, 1994. Section 7.2.
- [3] I.A. Hiskens and J. Alseddiqui. Sensitivity, Approximation, and Uncertainty in Power System Dynamic Simulation. *Power Systems, IEEE Transactions on*, 21(4):1808-1820, November 2006.
- [4] I.A. Hiskens and P.J. Sokolowski. Systematic Modeling and Symbolically Assisted Simulation of Power Systems. *Power Systems, IEEE Transactions on*, 16(2):229-234, May 2001.
- [5] C. Duclut, S. Backhaus, and M. Chertkov. Hysteresis, Phase Transitions and Dangerous Transients in Electrical Power Distribution Systems. *Physical Review E*, 87(6):062802-62818, June 2013.
- [6] I.A. Hiskens and M.A. Pai. Trajectory Sensitivity Analysis of Hybrid Systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 47(2):204-220, February 2000.
- [7] V. Donde and I.A. Hiskens. Dynamic Performance Assessment: Grazing and Related Phenomena. *IEEE Transactions on Power Systems*, 20(4):1967-1975, November 2005.
- [8] V. Donde and I.A. Hiskens. Shooting Methods for Locating Grazing Phenomena in Hybrid Systems. *International Journal of Bifurcation and Chaos*, 16(3):671-692, 2006.