

# An Iterative Approach to Calculating Dynamic ATC

I.A. Hiskens

Department of Electrical and Computer Engineering  
The University of Newcastle  
Callaghan NSW 2308  
Australia

M.A. Pai

P.W. Sauer

Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign  
1406 W. Green Street  
Urbana IL 61801  
USA

*Abstract*— Stability limits place restrictions on the available transfer capability (ATC) of power systems. Calculation of these limits is therefore very important, but has traditionally been quite difficult. This paper proposes an iterative algorithm for determining parameter values which result in marginal stability of a system. (A system is marginally stable for a particular disturbance if the post-disturbance trajectory lies on the stability boundary.) A knowledge of the critical parameter values allows the dynamic ATC to be determined. The algorithm is based on the Gauss-Newton solution of a nonlinear least-squares problem. This solution process uses trajectory sensitivities.

## I. INTRODUCTION

IN many power systems, the maximum power transfer across the network is limited by stability considerations. If the transfer level is raised too high, instability may occur for certain disturbances. To minimize this risk, operators are guided by constraints which ensure that if certain prescribed large disturbances occur, for example faults or generator tripping, then the system will continue to operate in a controlled fashion. A good knowledge of these stability constraints is very important. If the constraints are underestimated, then systems tend to be operated over-conservatively, with an associated economic penalty. However should the constraints be overestimated, then the system may well be operated in a vulnerable state without a clear understanding of the risk. Unfortunately though, the computation of these stability limits, and hence the dynamic available transfer capability (ATC), is difficult.

Underlying the calculation of dynamic ATC is the need to find a loading condition (set of parameters) at which the disturbed system is marginally stable. This set of parameters may include the generation schedule, loads, and controller setpoints, for example. Generally it is necessary to consider many disturbance scenarios, with a critical loading condition being calculated for each case. A comparison of those critical loading conditions with the current operating point gives the desired dynamic ATC.

The concept is straightforward, but implementation is difficult. It is not easy to find a marginally stable system. Lyapunov-based direct methods provide a useful approach. However modelling restrictions limit their application and accuracy. Otherwise, no systematic approach has been available. Hence stability limits are generally determined off-line, for a limited set of operating conditions and disturbance scenarios. The results of these studies are then used to guide operators.

This paper proposes a novel approach to finding parameters that correspond to marginally stable system behaviour.

It is shown in the sequel that the proposed algorithm is applicable (under fairly mild assumptions) to any system which can be modelled as a set of differential-algebraic-discrete (DAD) equations. The DAD model is extremely general, and allows full representation of nonlinear/nonsmooth devices. Examples include higher order machine models, controller limits and arbitrarily complicated protection characteristics. This model is discussed in Section II, along with some useful background system theory, and an overview of trajectory sensitivities. The proposed algorithm is described in Section III, and an example is explored in Section IV. Conclusions are given in Section V.

## II. BACKGROUND

### A. Model

Power systems frequently exhibit a mix of continuous time dynamics, discrete-time and discrete-event dynamics, switching action and jump phenomena. It is shown in [1] that such systems, known generically as *hybrid systems*, can be modelled by a set of switched differential-algebraic equations, coupled with equations to describe state resetting. A compact version of that differential-algebraic-discrete (DAD) model can be written as

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

$$0 = g^{(0)}(\underline{x}, y) \quad (2)$$

$$0 = \begin{cases} g^{(i-)}(\underline{x}, y) & y_{d,i} < 0 \\ g^{(i+)}(\underline{x}, y) & y_{d,i} > 0 \end{cases} \quad i = 1, \dots, d \quad (3)$$

$$\underline{x}^+ = \underline{h}_j(\underline{x}^-, y^-) \quad y_{e,j} = 0 \quad j \in \{1, \dots, e\} \quad (4)$$

where

$$\underline{x} = \begin{bmatrix} x \\ z \\ \lambda \end{bmatrix}, \quad \underline{f} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}, \quad \underline{h}_j = \begin{bmatrix} x \\ h_j \\ \lambda \end{bmatrix}.$$

An expanded presentation of this model is given in [1]. However this compact version is convenient for the later algorithm development. In this model,

- $x$  are the continuous dynamic states, for example generator angles, velocities and fluxes;
- $z$  are discrete dynamic states, such as transformer tap positions and protection relay logic states;

- $y$  are algebraic states, e.g., load bus voltage magnitudes and angles; and
- $\lambda$  are parameters such as generator mechanical powers, line reactances and switching times.

The function  $\underline{f}$  is structured so that, of the dynamic states  $\underline{x}$ , only  $x$  evolves continuously over time. Similarly, the state reset equations  $\underline{h}_j$  ensure that only the discrete states  $z$  undergo a step change at reset events. It follows that the parameters  $\lambda$  remain constant over all time. The algebraic states  $y$  generally vary continuously as  $x$  evolves. They may take a step change at reset events when  $z$  changes, and at constraint switching events.

Away from discontinuities, system dynamics evolve continuously according to the familiar differential-algebraic (DA) model

$$\dot{\underline{x}} = \underline{f}(\underline{x}, y) \quad (5)$$

$$0 = g(\underline{x}, y). \quad (6)$$

where  $g$  is composed of (2) together with functions from (3) chosen depending on the signs of the elements of  $y_d$ .

The system *flows* for  $x$  and  $y$  are defined as

$$\phi(\underline{x}_0, t) = \begin{bmatrix} \phi_{\underline{x}}(\underline{x}_0, t) \\ \phi_y(\underline{x}_0, t) \end{bmatrix} = \begin{bmatrix} \underline{x}(t) \\ y(t) \end{bmatrix} \quad (7)$$

where  $\underline{x}(t)$  and  $y(t)$  satisfy (1)-(4), along with initial conditions,

$$\phi_{\underline{x}}(\underline{x}_0, t_0) = \underline{x}_0 \quad (8)$$

$$g(\underline{x}_0, \phi_y(\underline{x}_0, t_0)) = 0. \quad (9)$$

## B. System theory

### B.1 Marginally stable trajectories

The proposed algorithm is based on the assumption that the limit set for the system is composed only of isolated points. (The limit set is the set of points to which the system converges in either forward or backward time.) This assumption excludes the existence of limit cycles, or other more complicated limit behaviour. It is also assumed that the trajectory and the stability boundary intersect, rather than the trajectory jumping the stability boundary, or vice versa, under the action of a switching or reset event.

Under those assumptions, the stability boundary is composed of the stable manifolds of type-1 unstable equilibrium points (UEPs) that lie on the boundary [2]. Therefore an unstable trajectory must pass through the stable manifold of a type-1 UEP. If a disturbance to the system is critically cleared, the trajectory will lie on the stable manifold of a UEP, and approach that UEP in infinite time. A trajectory which is nearly critically cleared, whether stable or unstable, will pass close to that UEP. But generally the UEP is unknown. Energy function methods require a knowledge of this ‘controlling UEP’ [3]. The proposed algorithm does not.

Equilibria of the system (5)-(6) satisfy

$$f(\underline{x}, y) = f(x, y, z; \lambda) = 0 \quad (10)$$

with (6) being satisfied everywhere. Therefore proximity to equilibria can be established using the quadratic cost function

$$\mathcal{J}(\underline{x}, y) = \frac{1}{2} f(\underline{x}, y)^t W f(\underline{x}, y) \quad (11)$$

where  $W$  is a diagonal matrix that is used to weight the relative importance of the various  $f$  functions. Near an equilibrium point, the cost  $\mathcal{J}(\underline{x}, y)$  becomes small. By monitoring  $\mathcal{J}(\underline{x}, y)$  along a trajectory, points that are local minima can be found. Many of these points indicate proximity to equilibria.

The proposed algorithm is based on the idea that  $\mathcal{J}(\underline{x}, y)$  defines a hypersurface in the variables  $t$  (time) and  $\underline{x}_0$  (initial conditions and parameters). Starting from a local minimum in  $\mathcal{J}$  on the initial trajectory,  $t$  and  $\underline{x}_0$  can be varied to minimize  $\mathcal{J}$  over that hypersurface. As  $\mathcal{J}$  is minimized, the minimum point moves closer towards a UEP, and hence the trajectory moves closer to passing through that UEP. More complete details are given in Section III. The example of Section IV illustrates these concepts.

It remains to determine the number of parameters that should be varied to move a trajectory so that it becomes coincident with the stability boundary. This is addressed in the following subsection.

### B.2 Parameter dimension

Let  $x \in \mathbb{R}^n$ , so that state-space has dimension  $n$ . (The algebraic states  $y$  can be thought of as implicit functions of  $x$ , and so do not influence the state-space dimension.) Then the stable manifold of a type-1 UEP has dimension  $n - 1$ . (This manifold can be thought of as being defined by a single equation in  $n$  unknowns.) If  $p$  parameters are allowed to vary then the stable manifold, denoted  $\mathcal{S}(x, \lambda)$ , becomes an  $(n + p - 1)$ -manifold in  $(n + p)$ -space.

Now consider the system flow  $\phi(\underline{x}_0, t)$ . For constant parameters,  $\phi(\underline{x}_0, t)$  is a 1-manifold, parameterized by time, that exists in state-space. The dimension of this manifold increases by one for each free parameter. Define  $\mathcal{F}(x, \lambda)$  as the  $(p + 1)$ -manifold, in  $(n + p)$ -space, that corresponds to  $p$  free parameters.

The intersection  $\mathcal{S}(x, \lambda) \cap \mathcal{F}(x, \lambda)$  is therefore generically a  $p$ -manifold in  $(n + p)$ -space [4]. Now we desire this intersection to be a trajectory that lies on the stability boundary. So the intersection must be a 1-manifold, i.e.,  $p = 1$ . Hence a single parameter should be varied in order to obtain a marginally stable system.

Two comments are in order:

- There is no guarantee that  $\mathcal{S}$  and  $\mathcal{F}$  will intersect for all choices of the single free parameter. However if they do intersect, then generically the intersection will be a single isolated trajectory that approaches the UEP.
- If more than one parameter are varied, then generically a continuum of trajectories will be obtained.

### C. Trajectory sensitivities

Trajectory sensitivities provide a way of quantifying the variation of a trajectory due to small changes in parameters

and/or initial conditions [5]. This section summarizes the main concepts. Further details can be found in [1].

To obtain the sensitivity of the flows  $\phi_{\underline{x}}$  and  $\phi_y$  to initial conditions, and hence to parameter variations, we form the Taylor series expansions of (7). Neglecting higher order terms gives

$$\Delta \underline{x}(t) = \frac{\partial \underline{x}(t)}{\partial \underline{x}_0} \Delta \underline{x}_0 \equiv \underline{x}_{x_0}(t) \Delta \underline{x}_0 \quad (12)$$

$$\Delta y(t) = \frac{\partial y(t)}{\partial \underline{x}_0} \Delta \underline{x}_0 \equiv y_{x_0}(t) \Delta \underline{x}_0. \quad (13)$$

It is important to keep in mind that  $\underline{x}_0$  incorporates  $\lambda$ , so sensitivity to  $\underline{x}_0$  includes sensitivity to  $\lambda$ . Equations (12),(13) provide the changes  $\Delta \underline{x}(t)$  and  $\Delta y(t)$  in a trajectory, at time  $t$ , for a given (small) change in initial conditions  $\Delta \underline{x}_0 = [\Delta x_0^t \ \Delta z_0^t \ \Delta \lambda^t]^t$ .

Full details of trajectory sensitivity calculations for the DAD model (1)-(4) are given in [1]. An important conclusion is that trajectory sensitivities are defined for complex event driven behaviour. It is shown in [1] that the sensitivities can be calculated efficiently as a by-product of using an implicit numerical integration technique, such as trapezoidal integration, to produce the nominal system trajectory.

Other recent power system applications of trajectory sensitivities include [6], [7], [8], [9], [10].

### III. ALGORITHM

#### A. Development

The algorithm is based on minimizing the quadratic cost function  $\mathcal{J}(\underline{x}, y)$  given by (11). But notice from (7) that  $\underline{x}$  and  $y$  are functions of  $\underline{x}_0$  and  $t$ . It was shown in Section II.B.2 that this optimization problem had isolated solutions if a single parameter was varied. Therefore we parameterize  $\underline{x}_0$  (the initial conditions and system parameters) by a scalar parameter  $\alpha$ . A discussion of this parameterization is given in Section III.B.

Therefore  $\mathcal{J}$  can be reformulated as

$$\begin{aligned} \mathcal{J}(t, \alpha) &= \frac{1}{2} f(\phi_{\underline{x}}(\underline{x}_0(\alpha), t), \phi_y(\underline{x}_0(\alpha), t))^t W \cdot \\ &\quad f(\phi_{\underline{x}}(\underline{x}_0(\alpha), t), \phi_y(\underline{x}_0(\alpha), t)) \\ &= \frac{1}{2} \tilde{f}(t, \alpha)^t W \tilde{f}(t, \alpha). \end{aligned} \quad (14)$$

The aim of the algorithm is to find  $t$  and  $\alpha$  that minimize (14).

Let  $z = [t \ \alpha]^t$ . Then (14) can be rewritten

$$\mathcal{J}(z) = \frac{1}{2} \tilde{f}(z)^t W \tilde{f}(z). \quad (15)$$

This unconstrained minimization can be achieved using a Gauss-Newton iterative method [11]. Iterations follow the scheme

$$z^{k+1} = z^k - \left( \nabla \tilde{f}(z^k)^t W \nabla \tilde{f}(z^k) \right)^{-1} \nabla \tilde{f}(z^k)^t W \tilde{f}(z^k) a \quad (16)$$

where  $a$  is an acceleration factor,

$$\nabla \tilde{f} = \begin{bmatrix} \frac{\partial \tilde{f}}{\partial t} & \frac{\partial \tilde{f}}{\partial \alpha} \end{bmatrix} \quad (17)$$

and

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial t} &= \frac{\partial f}{\partial \underline{x}} \frac{\partial \phi_{\underline{x}}}{\partial t} + \frac{\partial f}{\partial y} \frac{\partial \phi_y}{\partial t} \\ &= f_x \dot{\underline{x}} + f_y \dot{y} \\ \frac{\partial \tilde{f}}{\partial \alpha} &= \frac{\partial f}{\partial \underline{x}} \frac{\partial \phi_{\underline{x}}}{\partial \alpha} \frac{\partial \underline{x}_0}{\partial \alpha} + \frac{\partial f}{\partial y} \frac{\partial \phi_y}{\partial \alpha} \frac{\partial \underline{x}_0}{\partial \alpha} \\ &= \left( f_x \underline{x}_{x_0} + f_y y_{x_0} \right) \frac{\partial \underline{x}_0}{\partial \alpha}. \end{aligned} \quad (18)$$

The notation  $\underline{x}_{x_0}$  and  $y_{x_0}$  refers to the trajectory sensitivities; see (12),(13). The vector  $\frac{\partial \underline{x}_0}{\partial \alpha}$  is discussed in Section III.B. Recall that the algebraic equations (6) are satisfied everywhere. From (6) we obtain

$$\begin{aligned} g_x \dot{\underline{x}} + g_y \dot{y} &= 0 \\ \Rightarrow \dot{y} &= -(g_y)^{-1} g_x \dot{\underline{x}}. \end{aligned}$$

Substituting back into (18) gives

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial t} &= (f_x - f_y (g_y)^{-1} g_x) \dot{\underline{x}} \\ &= (f_x - f_y (g_y)^{-1} g_x) \underline{f}(\underline{x}, y). \end{aligned} \quad (20)$$

Iterations of the Gauss-Newton method cease when either  $\tilde{f}(z^k) \approx 0$  or when  $\nabla \tilde{f}(z^k)^t \tilde{f}(z^k) \approx 0$ . The former case corresponds to an equilibrium point lying on the trajectory. If that point is not the stable equilibrium point (SEP), then it means a marginally stable trajectory has been found, i.e., parameter values which cause the trajectory to run to a UEP have been identified. (Note that because a UEP can only be approached in infinite time, the optimal value of  $t$  given by the Gauss-Newton method will theoretically approach infinity. However in practice, iterations can be halted before  $t$  grows too large.)

The second condition corresponds to some local optimum point where  $\tilde{f}$  is orthogonal to the space spanned by  $\nabla \tilde{f}$ . It is not clear that such points have any physical significance.

#### B. Parameterization of $\underline{x}_0$

The scalar parameter  $\alpha$  describes the way in which  $\underline{x}_0$ , the initial conditions and system parameters, are to be varied in order to find a marginally stable system trajectory. A single element of  $\underline{x}_0$  may be varied independently, in which case  $\frac{\partial \underline{x}_0}{\partial \alpha}$  is a vector of zeros except for a single nonzero entry in the appropriate location. Alternatively  $\alpha$  may influence a few elements of  $\underline{x}_0$ . Then  $\frac{\partial \underline{x}_0}{\partial \alpha}$  gives a direction vector in  $\underline{x}_0$ -space.

Typically, in studies of system behaviour, the system starts from a stable equilibrium point. Then variation of  $\alpha$  may well shift the equilibrium point. From (5),(6), and

using the fact that  $\underline{f} = [f^t \ 0^t \ 0^t]^t$ , the starting equilibrium point is described by

$$\begin{aligned} 0 &= f(\underline{x}_0(\alpha), y_0) \\ 0 &= g(\underline{x}_0(\alpha), y_0). \end{aligned}$$

Differentiating gives

$$\begin{aligned} 0 &= f_x \frac{\partial \underline{x}_0}{\partial \alpha} d\alpha + f_y dy_0 \\ 0 &= g_x \frac{\partial \underline{x}_0}{\partial \alpha} d\alpha + g_y dy_0 \end{aligned}$$

and further manipulation yields

$$0 = (f_x - f_y(g_y)^{-1}g_x) \frac{\partial \underline{x}_0}{\partial \alpha} = f_x^* \frac{\partial \underline{x}_0}{\partial \alpha}. \quad (21)$$

Note that  $f_x^*$  is not square. With  $x = [x^t \ z^t \ \lambda^t]^t \in \mathbb{R}^{n+l+p}$ ,  $f_x^*$  is an  $n \times (n+l+p)$  matrix. Equation (21) can be rewritten

$$0 = f_x^* \frac{\partial x_0}{\partial \alpha} + f_z^* \frac{\partial z_0}{\partial \alpha} + f_\lambda^* \frac{\partial \lambda}{\partial \alpha}.$$

The  $x_0$  are dependent variables, whilst  $z_0$  and  $\lambda$  are independent variables. The desired  $\frac{\partial x_0}{\partial \alpha}$  can be found using

$$\frac{\partial x_0}{\partial \alpha} = -(f_x^*)^{-1} \left( f_z^* \frac{\partial z_0}{\partial \alpha} + f_\lambda^* \frac{\partial \lambda}{\partial \alpha} \right)$$

where  $\frac{\partial z_0}{\partial \alpha}$  and  $\frac{\partial \lambda}{\partial \alpha}$  are independent, and hence specified.

Note that for some choices of  $\alpha$ , such as fault clearing time or machine inertia,  $\frac{\partial x_0}{\partial \alpha}$  will be zero. But for choices such as load level or generator mechanical torque,  $\frac{\partial x_0}{\partial \alpha}$  will reflect the change in the equilibrium point.

#### IV. EXAMPLE

The algorithm of Section III is illustrated using a power system of the form shown in Figure 1. Both generators were represented by a two axis (fourth order) machine model [12]. In all cases, the system was disturbed by the application of a three phase fault at the terminal bus of Generator 1.

Two studies are presented. In the first, the calculation of the critical fault clearing time is considered. The second study, which is more relevant to dynamic ATC, explores the maximum loading of the generators.

##### A. Critical clearing time

For this study, the fault-on time  $t_f$  was chosen as the unknown parameter  $\alpha$  which was varied to minimize the cost function  $\mathcal{J}$  given by (14). The fault switching mechanism, and hence  $t_f$ , was incorporated into the DAD model (1)-(4), by introducing the equations

$$\dot{x}_t = 1$$

$$0 = (x_t - t_{init})(x_t - (t_{init} + t_f)) - y$$

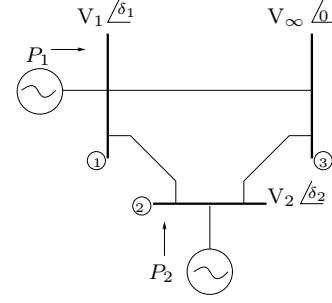


Fig. 1. Two machine infinite bus system.

where  $x_t$  is a timer, and  $t_{init}$  is the time at which the fault is initiated. The algebraic variable  $y$  is negative during the fault-on period, and positive otherwise. It is used to switch an algebraic equation, as in (3). When  $y$  is negative, fault current is added to the net current injection at the faulted bus. Therefore  $t_f$  can be treated just like any other parameter. Variation of  $t_f$  does not affect other initial conditions though, so  $\frac{\partial x_0}{\partial \alpha}$  in (19) and (21) is a vector of zeros except for a 1 in the location corresponding to  $t_f$ .

In this investigation an initial guess of  $t_f = 0.40$ sec was used. It can be seen from the phase portrait of Figure 2 that the system was clearly unstable. Subsequent Gauss-Newton iterations using (16) are given in Table I. The final estimate of  $t_f = 0.2931$ sec compares well with the actual critical value of  $t_f^* = 0.2866$ sec obtained by repetitive simulation. The iterations are illustrated in Figure 2. (Note that this figure is only a two dimensional projection of the eight dimensional state-space.) The figure shows the trajectories corresponding to the initial guess, the iterations and final estimate, and the actual critical value.

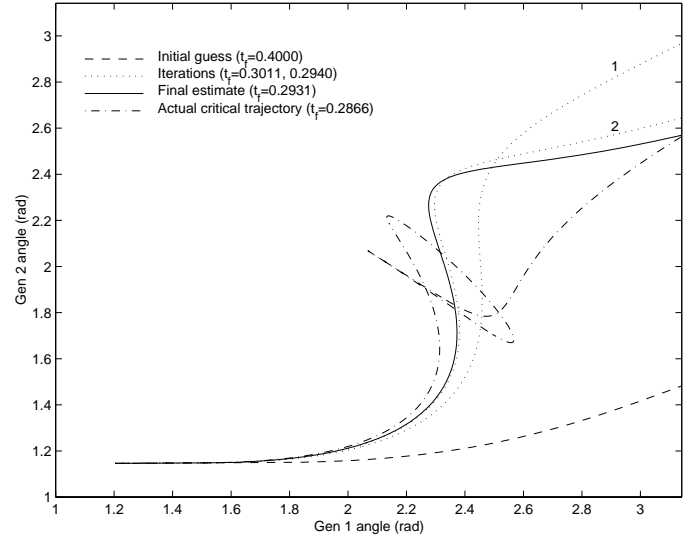


Fig. 2. Phase portrait view of trajectories, study A.

Figure 2 shows that behaviour becomes rather complicated as the critical parameter value is approached. This reflects into the cost function  $\mathcal{J}$  as multiple local minima. In this case the algorithm converged to a nearby local min-

Initial guess (sec)	Iterations (sec)			Actual value (sec)
	1	2	3	
0.4000	0.3011	0.2940	0.2931	0.2866

TABLE I  
ESTIMATES OF  $t_f$ .

imum rather than the desired minimum corresponding to the UEP. However from a practical perspective, the error in the final estimate of  $t_f$  is insignificant. If desired, an improved final estimate can be obtained by restarting the iterative solution process.

Further illustration is provided by Figure 3, where the cost function  $\mathcal{J}$  is plotted against time. It is clear that for the initial parameter guess,  $\mathcal{J}$  had only a single minimum, near 1sec. However as the parameter estimate was progressively improved, a local minimum developed near 1.5sec. The solution process in this case locked onto that local minimum. By restarting the process from the deeper minimum at 2.4sec, an improved estimate of  $t_f = 0.2878$ sec was obtained.

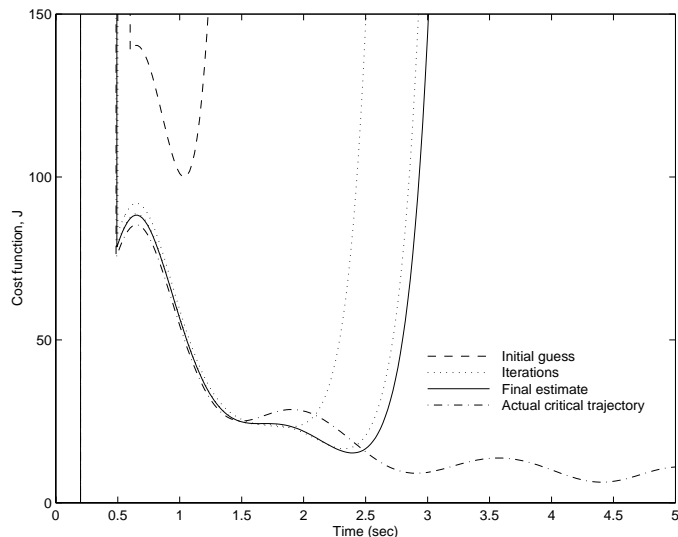


Fig. 3. Cost function  $\mathcal{J}$  versus time.

Throughout this sequence of studies an acceleration factor of  $a = 1$  was used. A more sophisticated choice of  $a$  may well have resulted in direct convergence to the improved estimate of  $t_f$ . Further investigations are required.

### B. Critical generator loading

The aim of this study was to determine the values of generator mechanical torques  $T_m$  which corresponded to the system being marginally stable, given the fault scenario described earlier. In this case the fault-on time was held fixed at 0.3sec. The study therefore effectively determined the maximum power that could be delivered by the generators whilst still ensuring stability for the specified fault condition. A comparison of this stability limited maximum power transfer with the nominal transfer level would yield the dynamic ATC.

An initial guess of  $T_m = 0.72$ pu was used for both generators. Also,  $T_m$  was varied in unison at the two generators during the iterations. (This was not a limitation of the algorithm, but rather was done for presentation clarity.) The phase portrait of Figure 4 shows the system response for the base case. The system was clearly stable. The Gauss-Newton algorithm proceeded to estimate a critical value of  $T_m = 0.7934$ pu. Repetitive simulations indicated that the actual critical value (to four decimal places) was  $T_m = 0.7930$ pu. Figure 4 shows the trajectories corresponding to the estimated and actual critical values. There is clear propinquity.

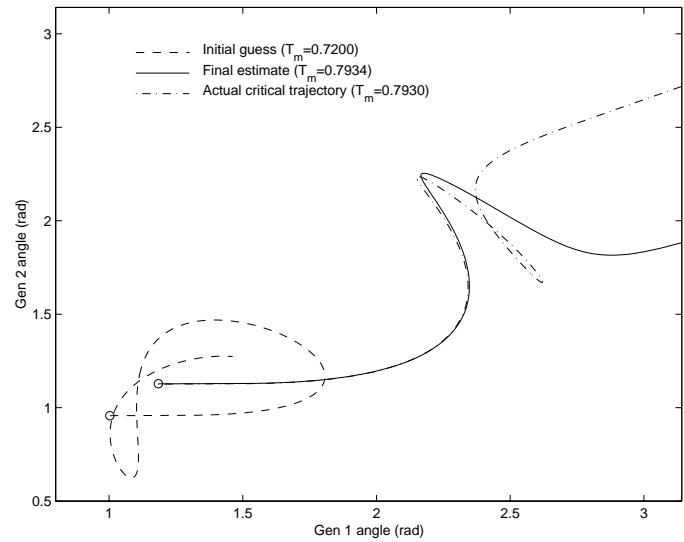


Fig. 4. Phase portrait view of trajectories, study B.

Notice in Figure 4 that the trajectories for  $T_m = 0.72$ pu and  $T_m = 0.7934$ pu start at different points. (The starting points are marked by a ‘o’.) This occurs because variation of  $T_m$  alters the operating point, i.e., the initial point  $\underline{x}_0$ . The dependence of  $\underline{x}_0$  on the parameter is reflected through  $\frac{\partial \underline{x}_0}{\partial \alpha}$ , in accordance with (21).

Convergence from the initial guess for  $T_m$  to the final estimate took 11 iterations. The iterations are shown by ‘x’ in the contour plot of Figure 5. This plot shows contours of the cost function  $\mathcal{J}$  for various values of  $T_m$  and time  $t$ . (Recall from (14) that  $\mathcal{J}$  is a function of  $t$  and  $\alpha$ , in this case  $T_m$ .) Progress is slow over the last few iterations because the cost function near the critical parameter value is quite flat. An improved strategy for choosing the acceleration factor would speed convergence.

It is interesting to note that the first iteration step is very small. This is because the cost function is quite flat near the initial parameter guess. In fact that initial point is near a saddle in  $\mathcal{J}$ . If the algorithm was started from a guess of  $T_m \approx 0.7$ pu, subsequent iterations would move away from the desired minimum, towards a minimum corresponding to a stable equilibrium point.

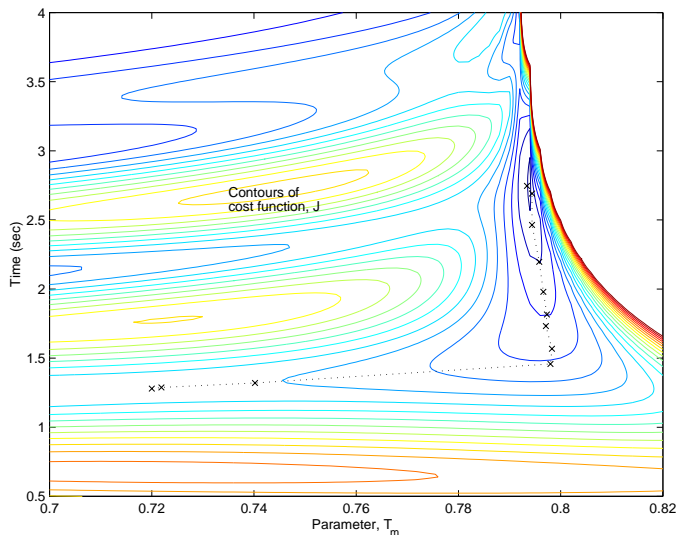


Fig. 5. Contours of cost function  $\mathcal{J}$  versus  $T_m$  and  $t$ .

## V. CONCLUSIONS

The paper develops an iterative approach to finding parameter values that result in marginally stable system behaviour for a specified disturbance. This approach is based on the fact that trajectories which lie on the stability boundary must approach an unstable equilibrium point. The algorithm varies parameters to minimize a cost function which is a measure of the distance from equilibria. By minimizing this cost function, trajectories move closer to passing through a UEP, and hence move closer to lying on the stability boundary.

Two studies have illustrated the algorithm. In the first, the fault-on time was chosen as the parameter of interest. The second study found the generator loading condition that corresponded to the transient stability limit. Calculation of the dynamic ‘available transfer capability’ follows directly from a knowledge of this maximal loading condition.

## ACKNOWLEDGMENTS

The authors acknowledge the support of the Australian Research Council through the project grant “Analysis and Assessment of Voltage Collapse”, the National Science Foundation through its grant NSF ECS 95-22547, and the Grainger Foundation.

## REFERENCES

- [1] I.A. Hiskens and M.A. Pai, “Trajectory sensitivity analysis of hybrid systems,” *IEEE Transactions on Circuits and Systems I*, Submitted for publication.
- [2] H-D. Chiang, M.W. Hirsch, and F.F. Wu, “Stability regions of nonlinear autonomous dynamical systems,” *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 16–27, January 1988.
- [3] M.A. Pai, *Energy Function Analysis for Power System Stability*, Kluwer Academic Publishers, Boston, MA, 1989.
- [4] V.W. Guillemin and A. Pollack, *Differential Topology*, Prentice Hall, Englewood Cliffs, NJ, 1974.
- [5] P.M. Frank, *Introduction to System Sensitivity Theory*, Academic Press, New York, 1978.
- [6] S.M. Benchluch and J.H. Chow, “A trajectory sensitivity method for the identification of nonlinear excitation system mod-

els,” *IEEE Transactions on Energy Conversion*, vol. 8, no. 2, pp. 159–164, June 1993.

- [7] M.J. Laufenberg and M.A. Pai, “A new approach to dynamic security assessment using trajectory sensitivities,” *IEEE Transactions on Power Systems*, To appear.
- [8] R.J. Galarza, J.H. Chow, W.W. Price, A.W. Hargrave, and P.M. Hirsch, “Aggregation of exciter models for constructing power system dynamic equivalents,” *IEEE Transactions on Power Systems*, To appear.
- [9] I.A. Hiskens and M. Akke, “Analysis of the Nordel power grid disturbance of January 1, 1997 using trajectory sensitivities,” *IEEE Transactions on Power Systems*, To appear.
- [10] Q. Wu, “Tap changing dynamic modeling and its effects on power system voltage behaviour,” Master of Engineering Thesis, Department of Electrical Engineering, University of Sydney, Australia, 1998.
- [11] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.
- [12] P.W. Sauer and M.A. Pai, *Power System Dynamics and Stability*, Prentice Hall, Upper Saddle River, NJ, 1998.