ECE 522
Power Systems Analysis II

3 – Power System Stability

Spring 2018
Instructor: Kai Sun
Outline

3.1 Transient stability
• Time-domain simulation
  – Explicit and implicit numerical integration techniques
  – Simulating a multi-machine system
• Transient stability analysis using direct methods
  – Transient stability of an SMIB system
  – SMIB equivalent based methods
  – Transient energy function based direct methods

3.2 Small signal stability

3.3 Voltage Stability
Transient Stability

• The ability of the power system to maintain synchronism when subjected to a severe disturbance such as a fault on transmission facilities, loss of generation or loss of a large load.
  – The system response to such disturbances involves large excursions of generator rotor angles, power flows, bus voltages, and other system variables.
  – Stability is influenced by the nonlinear characteristics of the system.
  – If the resulting angular separation between the machines in the system remains within certain bounds, the system maintains synchronism.
  – If loss of synchronism due to transient instability occurs, it will usually be evident within 2-3 seconds after the initial disturbances.
Methods for Transient Stability Analysis

Analyzing the stability of a nonlinear system following a disturbance

- **Time-domain simulation** solves the nonlinear differential-algebraic equations (DAEs) with known initial values $x=x_0$ and $t=t_0$ by using step-by-step numerical integration (explicit or implicit).

  $\dot{x} = f(x, u)$ \hspace{1cm} DE

  $0 = g(x, u)$ \hspace{1cm} AE

- **Direct methods**, based on Lyapunov’s second method, determine stability without explicitly solving the DAEs:

  1. Define a Transient Energy Function (TEF) as a possible/approximate Lyapunov function $V(x)$
  2. Compare the TEF to a critical energy $V_{cr}$ to judge stability
3.1.1 Time-Domain Simulation
Numerical Integration Methods

• The differential equations to be solved for stability analysis are nonlinear ODEs (ordinary differential equations) with known initial values $x=x_0$ and $t=t_0$

$$\frac{dx}{dt} = f(x,t)$$

where $x$ is the state vector of $n$ dependent variables and $t$ is the independent variable (time). Our objective is to solve $x$ as a function of $t$

• Explicit Methods
  – computing the value of $x$ at any time $t$ using the values of $x$ from only the previous time steps, e.g. Euler and R-K methods

  Euler method: $x_{i+1} = x_i + f(x_i, t_i)\Delta t$

• Implicit Methods
  – Using interpolation functions involving future time steps for the expression under the integral, e.g. the Backward Euler and Trapezoidal Rule methods

  Backward Euler method: $x_{i+1} = x_i + f(x_{i+1}, t_{i+1})\Delta t$
Euler Method

- The Euler method is equivalent to using the first two terms of the Taylor series about $x$ around the point $(x_0, t_0)$, referred to as a **first-order method** (error is on the order of $\Delta t^2$)
  - Approximate the curve at $x=x_0$ and $t=t_0$ by its tangent

\[
\frac{dx}{dt}\bigg|_{x_0} = f(x_0, t_0) \quad \Delta x \approx \frac{dx}{dt}\bigg|_{x_0} \Delta t
\]

\[
x_1 = x_0 + \Delta x = x_0 + \frac{dx}{dt}\bigg|_{x_0} \Delta t = x_0 + f(x_0, t_0) \Delta t
\]

- At step $i+1$:

\[
x_{i+1} = x_i + f(x_i, t_i) \Delta t
\]

- It is explicit compared to Backward Euler Method (implicit)

\[
x_{i+1} = x_i + f(x_{i+1}, t_{i+1}) \Delta t
\]

- The standard Euler method results in inaccuracy because it uses the derivative only at the beginning of the interval as though it applied throughout the interval
Modified Euler (ME) Method

- Modified Euler method consists of two steps:
  
  (a) Predictor step:
  
  \[ x_1^p = x_0 + \left. \frac{dx}{dt} \right|_{x_0} \Delta t = x_0 + f(x_0, t_0)\Delta t \]

  Slope at the beginning of \( \Delta t \)
  
  The derivative at the end of the \( \Delta t \) is estimated using \( x_1^p \)

  \[ \left. \frac{dx}{dt} \right|_{x_1^p} \approx f(x_1^p, t_1) \quad \text{Estimated slope at the end of } \Delta t \]

  (b) Corrector step:

  \[ x_1^c = x_0 + \left. \frac{dx}{dt} \right|_{x_0} + \left. \frac{dx}{dt} \right|_{x_1^p} \Delta t = x_0 + \frac{f(x_0, t_0) + f(x_1^p, t_1)}{2} \Delta t \]

  \[ x_{i+1}^c = x_i + \frac{f(x_i, t_i) + f(x_{i+1}^p, t_{i+1})}{2} \Delta t \]

- It is a second-order method (error is on the order of \( \Delta t^3 \))

- Step size \( \Delta t \) must be small enough to obtain a reasonably accurate solution, but at the same time, large enough to avoid the numerical instability with the computer.
Runge-Kutta (R-K) Methods

- General formula of the 2\textsuperscript{nd} order R-K method (RK2):
  (error is on the order of $\Delta t^3$)
  \[
  k_1 = f(x_0, t_0) \\
  k_2 = f(x_0 + \alpha k_1, t_0 + \beta \Delta t) \\
  x_1 = x_0 + (a_1 k_1 + a_2 k_2) \Delta t
  \]

  At Step $i+1$:
  \[
  k_1 = f(x_i, t_i) \\
  k_2 = f(x_i + \alpha k_1, t_i + \beta \Delta t) \sim O(\Delta t) \\
  x_{i+1} = x_i + (a_1 k_1 + a_2 k_2) \Delta t \sim O(\Delta t^2)
  \]

- General formula of the 4\textsuperscript{th} order R-K method:
  (error is on the order of $\Delta t^5$)
  \[
  x_{i+1} = x_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \Delta t \sim O(\Delta t^4)
  \]

\[
\frac{dx}{dt} = f(x, t)
\]

When $a_1=a_2=1/2$, $\alpha=\beta=1$, the RK2 method becomes the ME method (i.e. a special case of RK2)
Numerical Stability of Explicit Integration Methods

• Numerical stability is related to the **stiffness** of the set of differential equations representing the system

• The **stiffness** is measured by the ratio of the largest to smallest time constant, or more precisely by \(|\lambda_{\text{max}}/\lambda_{\text{min}}|\) of the linearized system.

• **Stiffness** in a transient stability simulation increases with more details (more smaller time constants) being modeled.

• Explicit integration methods have weak stability numerically; with stiff systems, the solution “blows up” unless a small step size is used. Even after the fast modes die out, small time steps continue to be required to maintain numerical stability
Implicit Methods

- Implicit methods use interpolation functions for the expression under the integral. “Interpolation” implies the function must pass through the yet unknown points at $t_1$.
- A widely used implicit integration method is the Trapezoidal Rule method. It uses linear interpolation.
- The stiffness of the system being analyzed affects accuracy but not numerical stability. With larger time steps, high frequency modes and fast transients are filtered out, and the solutions for the slower modes is still accurate. For example, for the Trapezoidal rule, only dynamic modes faster than $f(x_n,t_n)$ and $f(x_{n+1},t_{n+1})$ are neglected.

\[ x_1 = x_0 + \int_{t_0}^{t_1} f(x,t) \, dt = x_0 + |A| + |B| \approx x_0 + |A| \]

\[ x_1 = x_0 + \frac{\Delta t}{2} \left[ f(x_0,t_0) + f(x_1,t_1) \right] \]

\[ x_{n+1} = x_n + \frac{\Delta t}{2} \left[ f(x_n,t_n) + f(x_{n+1},t_{n+1}) \right] \]

Compared to ME method:

\[ x_1 = x_0 + \frac{\Delta t}{2} \left[ f(x_0,t_0) + f(x_1^p,t_1) \right] \]
Comparison of Explicit and Implicit Methods

\[ \dot{x} = f(x, t) \approx \lambda_{\text{max}} x \]

### Euler Method (explicit)

\[
x_i = x_{i-1} + f(x_{i-1}, t_{i-1}) \Delta t
\]

\[
\approx x_{i-1} + \lambda_{\text{max}} x_{i-1} \Delta t
\]

\[
= x_{i-1} (1 + \lambda_{\text{max}} \Delta t)
\]

\[
x_i = x_0 (1 + \lambda_{\text{max}} \Delta t)^i
\]

The method is numerically stable if

\[
| 1 + \lambda_{\text{max}} \Delta t | < 1
\]

\[\Leftrightarrow \lambda_{\text{max}} \text{ has a negative real part and} \]

\[
\Delta t < \frac{2}{|\lambda_{\text{max}}|}
\]

### Backward Euler Method (implicit)

\[
x_i = x_{i-1} + f(x_i, t_i) \Delta t
\]

\[
\approx x_{i-1} + \lambda_{\text{max}} x_i \Delta t
\]

\[
x_i = x_{i-1} \frac{1}{1 - \lambda_{\text{max}} \Delta t}
\]

\[
x_i = x_0 \left( \frac{1}{1 - \lambda_{\text{max}} \Delta t} \right)^i
\]

\[\Delta t \text{ can be arbitrarily large as long as}\]

\[\lambda_{\text{max}} \text{ has a negative real part}\]

\[\text{(this method has A-Stability)}\]
Simulating a Multi-Machine System in a Simplified Model

• Consider these classic simplifying assumptions:
  – Each synchronous machine is represented by a voltage source $E'$ with constant magnitude $|E'|$ behind $X'd$ (neglecting armature resistances, the effect of saliency and the changes in flux linkages); The mechanical rotor angle of each machine coincides with the angle of $E'$
  – The governor’s actions are neglected and the input powers $P_{mi}$ are assumed to remain constant during the entire period of simulation
  – Using the pre-fault bus voltages, all loads are converted to equivalent admittances to ground. Those admittances are assumed to remain constant (constant impedance load models)
  – Damping or asynchronous powers are ignored.
  – Machines belonging to the same station swing together and are said to be coherent. A group of coherent machines is represented by one equivalent machine
• Solve the initial power flow and determine the initial bus voltage phasors $V_i$.
• Terminal currents $I_i$ of $m$ generators prior to disturbance are calculated by their terminal voltages $V_i$ and power outputs $S_i$, and then used to calculate $E'_i$

$$I_i = \frac{S_i^*}{V_i^*} = \frac{P_i - jQ_i}{V_i^*} \quad E'_i = V_i + jX'_d i I_i \quad i = 1, 2, \ldots, m$$

• All loads are converted to equivalent admittances:

$$y_{i0} = \frac{S_i^*}{|V_i|^2} = \frac{P_i - jQ_i}{|V_i|^2}$$

• To include voltages behind $X'_d i$, add $m$ internal generator buses to the $n$-bus power system network to form a $n+m$ bus network (ground as the reference for voltages):
• Node voltage equation with ground as reference

\[ \mathbf{I}_{bus} = \mathbf{Y}_{bus} \mathbf{V}_{bus} \]

\[
\begin{bmatrix}
I_1 \\
I_2 \\
\vdots \\
I_n \\
I_{n+1} \\
\vdots \\
I_{n+m}
\end{bmatrix}
= 
\begin{bmatrix}
Y_{11} & \cdots & Y_{1n} \\
Y_{21} & \cdots & Y_{2n} \\
\vdots & \cdots & \vdots \\
Y_{n1} & \cdots & Y_{nn}
\end{bmatrix}
\begin{bmatrix}
Y_{1(n+1)} & \cdots & Y_{1(n+m)} \\
Y_{2(n+1)} & \cdots & Y_{2(n+m)} \\
\vdots & \cdots & \vdots \\
Y_{n(n+1)} & \cdots & Y_{n(n+m)}
\end{bmatrix}
\begin{bmatrix}
V_1 \\
V_2 \\
\vdots \\
V_n \\
E'_{n+1} \\
\vdots \\
E'_{n+m}
\end{bmatrix}
\]

\( \mathbf{I}_{bus} \) is the vector of the injected bus currents

\( \mathbf{V}_{bus} \) is the vector of bus voltages measured from the reference node

\( \mathbf{Y}_{bus} \) is the bus admittance matrix:

- \( Y_{ii} \) (diagonal element) is the sum of admittances connected to bus \( i \)
- \( Y_{ij} \) (off-diagonal element) equals the negative of the admittance between buses \( i \) and \( j \)

Compared to the \( \mathbf{Y}_{bus} \) for power flow analysis, additional \( m \) internal generator nodes are added and \( Y_{ii} (i \leq n) \) is modified to include the load admittance at node \( i \)
• To simplify the analysis, all nodes other than the generator internal nodes are eliminated as follows

\[
\begin{bmatrix}
0 \\
I_m
\end{bmatrix} = 
\begin{bmatrix}
Y_{nn} & Y_{nm} \\
Y_{nm}^t & Y_{mm}
\end{bmatrix} 
\begin{bmatrix}
V_n \\
E_m'
\end{bmatrix}
\]

\[
V_n = -Y_{nn}^{-1}Y_{nm}E_m'
\]

\[
I_m = [Y_{mm} - Y_{nm}^t Y_{nn}^{-1} Y_{nm}] E_m'
= Y_{bus}^{red} E_m'
\]

\[
S_{ei} = E_i'^* I_i
\quad \text{where } I_i = \sum_{j=1}^{m} E_j' Y_{ij}
\]

\[
P_{ei} = \Re [E_i'^* I_i]
\]

\[
= \sum_{j=1}^{m} |E_i'| |E_j'| |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j)
\]

where \(\theta_{ij}\) is the angle of \(Y_{ij}\)

\[
0 = Y_{nn} V_n + Y_{nm} E_m'
\]

\[
I_m = Y_{nm}^t V_n + Y_{mm} E_m'
\]

\[
Y_{bus}^{red} = Y_{mm} - Y_{nm}^t Y_{nn}^{-1} Y_{nm}
\]

\(Y_{bus}^{red}\) needs to be updated whenever the network is changed.