21st Century Power System Dynamics

EECS 290

Spring 2019

Feb 25 2019 Lecture notes.

Today

1. Review project goals for each group

   Goal: Identify common objectives, gather class-wide feedback

2. Numerical integration and DAEs.

   Goal 1: Understand need for numerical integration and challenge posed by DAEs

   G2: Understand basic concept of numerical integration, issues of convergence and stability,

   G3: Understand specific case of implicit integration

3. Matlab example

   G1: Understand basic mechanics of using ode15s.

4. Electromagnetic transients

   G1: Undertand mathematical description relating magnetic field to fault currents.

   G2: Understand key time constants governing dynamics.

Group Project Review

1. Lin: (Sam, Ricky, Anna, Vaggelis)
   a. implement the model
   b. add line impedance
   c. modify converter → voltage source
   d. re-run analysis

2. Markovic: (Paty, Goran, Jose, Nathan)
   a. implement the model
   b. add synchronous machine
   c. Suggest: identify goal for analysis.
   d Suggest: examine converter model assumptions, time constants.

3. Curi (Keith, Jonathan, Victoria, Rodrigo)
   a. build simulink or Matlab simulation "Full order"
   b. simulation validation for reduced order.
   c Suggest: Higher order converter models (e.g. Markovic) to test assumptions

Ramasubramanian Rose, Jaimie, Jason, Phillippe

- Large scale modeling
- Custom faults
- Exploring ground truth?

Numerical integration and DAEs.

Goal 1: Understand basic concept of numerical integration, issues of convergence and stability,

G2: Understand challenge posed by DAEs

G3: Understand specific case of implicit integration

Numerical Integration: Introduction

Suppose you have

$$\dot{x} = f(x,t)$$

and f is not analytically solvable, ie we can't find

$$x = F(t)$$

what to do? we can approximate the solution numerically:

first,

$$t_{k+1} = t_k + \Delta t$$

Then,

w is a power series approximation of $f \Rightarrow$ something we can integrate.

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(x,t)\,dt \approx x_k + \int_{t_k}^{t_{k+1}} w(t)\,dt$$

Numerical integration introduction, continued

All methods use the basic principle of

1. approximating f
2. Iterating the solution to get points along a trajectory (rather than a continuous function).

General form:

$$x_{k+1} = x_k + \left(\sum_{j=1}^{r} b_j f_{k+1-j} + b_0 f_{k+1} \right)$$

requires estimate at x_{k+1}

$\Rightarrow$ "implicit" method.

$b_0 = 0 \Rightarrow$ "explicit" method

Numerical integration: First order, explicit

$$x_{k+1} = x_k + h f_k = x_0 + \sum_{i=0}^{k} h f_i \iff x(T) = x(0) + \int_0^T f \, dt$$

step size, $h = \Delta t$ when doing time integration

Small step size $\Rightarrow$ analogous to analytical integration

Higher order methods simply involve using higher order taylor expansions.

Numerical methods: first order, implicit

$$x_{k+1} = x_k + h f_{k+1}$$

Now we need to figure out f_{k+1}.

$$f_{k+1} = f(x_{k+1})$$

But we don't know x_{k+1} yet!

Solution: "functional iteration"

$$x_{k+1}^{(\ell+1)} = x_k + h f\left(x_{k+1}^{(\ell)}\right)$$

← Holding k fixed, we iterate on ℓ until convergence. Then we move on to the next k.

Numerical methods: first order, implicit, ctd

One can also use Newton's method to iterate faster.

If solving $F(x) = 0$, then

$$x^{(l+1)} = x^{(l)} - \left[\frac{\partial F}{\partial x}\right]_{(l)}^{-1} F(x^{(l)})$$

Jacobian of F.

Numerical Methods: Two Key Considerations

1. Error. This is an approximation! In general, higher order methods have less error, as do implicit methods. Reducing step size reduces error, at the expense of computing time.

2. Stability. $x_{k+1} = g(x_k)$ is a discrete dynamical system. It is stable if it remains in the vicinity of the exact solution. In general, implicit methods have greater stability. Reducing step size improves stability. Order of the method influences stability, but not systematically one way or the other

Numerical Methods : A third consideration

3. (Implicit methods only) Convergence.

will the sequence $x_{k+1}^{(l+1)} = x_k + hf(x_{k+1}^{(l)})$ converge?

I.e. will $x_{k+1}^{(l+1)} - x_{k+1}^{(l)} \to 0$?

The answer depends on

- Step size, h
- Properties of the Jacobian, $\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \\ \frac{\partial f_2}{\partial x_2} & \ddots & \\ & & \ddots \end{bmatrix}$

Numerical Integration: Good solvers do all the work for you.

Matlab and Simulink can adjust....

- Step length
- Order of the method

.... To balance stability, convergence, error, and computing time

I believe some solvers will also choose Adams vs. Gear, Explicit vs. implicit as well.

Numerical Methods: "Adams" vs. "Gear" Formulae

Note: Discussion to this point focuses on using a Taylor expansion of $f(x,t)$.

It's also possible (but in my experience, uncommon) to expand a formula for $x(t)$.

These are called "Gear" methods, and we won't cover them.

The methods we have covered (approximating $f(x,t)$) are known as "Adams" formulae.

Numerical Integration: What is special about Power Systems?

Network constraints are often represented as "algebraic constraints," e.g. network equations

bus voltage phasors

$$\mathbb{I} = \mathbb{Y}\mathbb{V}$$

vector of current injection phasos

admittance matrix (complex, not phasor entries)

$$\Rightarrow \mathbb{S} = \mathbb{V}\mathbb{I}^*$$

apparent power.

Though one could include electromagnetic line dynamics, these are usually omitted.

Numerical Integration : What is special about Power Systems, ctd

So we typically think of power system dynamics as :

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

$$0 = g(x, y) \quad \Leftarrow \text{contain the network equations}$$

x: dynamic variables, e.g. frequency, generator voltages, angles and so on.

y: static loads, network flows

This is an extreme case of what's known as a "stiff" dynamical system.

Stiff systems have one or more variables whose dynamics are significantly faster than the others. In this case the algebraic variables change <u>instantly</u>

Numerical Integration: What is special about Power Systems, ctd

Challenges:

① Need an intial network flow solution $\Rightarrow$ have to solve the "power flow" problem

② We can't solve algebraic eqns directly during simulation, esp. when $\exists$ static P, Q injections (i.e. load and gen are indep. of $\mathbb{V}$ and $\mathbb{I}$).

Solution: If you've learned about the power flow problem before, you know this is solved via numerical methods as well - in particular Newton's method.

But! We're already using Newton's method in implicit numerical integration - so this is a simple add-on.

Matlab: ode15s

- ode15s → "stiff"
  ode15s → variable order (1st to 5th, chooses adaptively)
  ode15s → numerically solves odes

- works by implicit numerical method.

- Important option: "mass matrix"

$$M \dot{x} = f(x)$$

  - Typically M is diagonal.
  - Diagonal elements weight the relative size of the rates of change
    → very different entries supports stiff systems
  - Zero entry ⇒ algebraic constraint.

Matlab Example

First order ODE:

```
function dydt = firstorder(y,t)
ẏ = 5y - 3
```

```
y(0) = 1
[t, y] = ode15s(@firstorder, tspan, y0)

plot(t, y)
```

Swing - ∞ - bus

```
function dydt = swing_one(y,t)

D = -0.1;

dydt =
```

Swing: two bus.

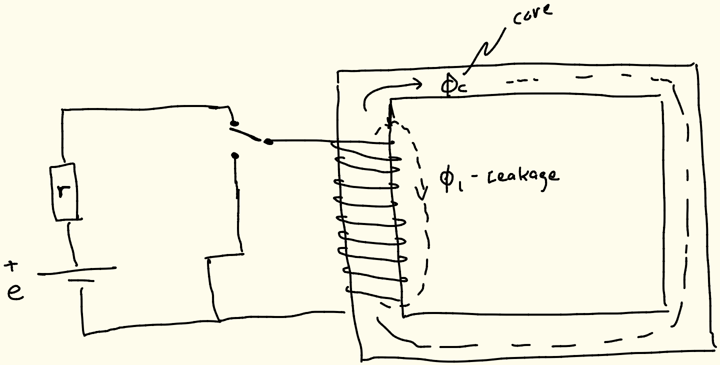
- Note singular mass matrix
- Note it solves the network first before integrating (e.g. if network flow does not match generator states)

Electromagnetic Transients

G1: Undertand mathematical description relating magnetic field to fault currents.

G2: Understand key time constants governing dynamics.

Electromagnetic transients



$$\phi = \phi_c + \phi_1$$

Ψ_o = flux linkage before switch

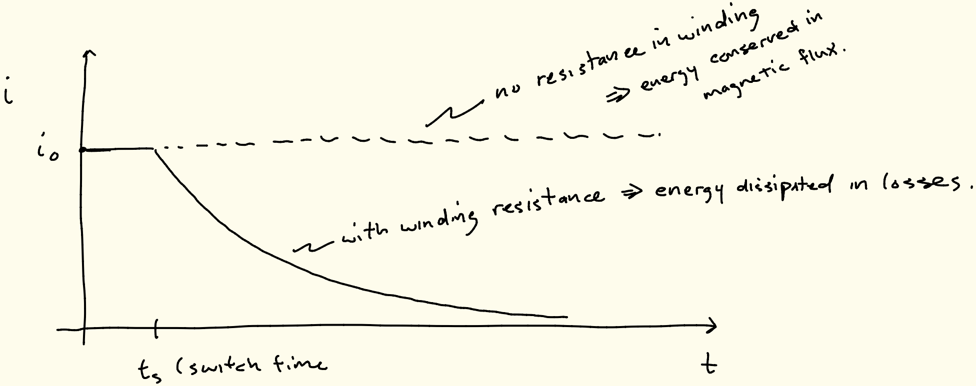
$= N\phi_o = L i_o$

initial current

turns

field strength in cross section area of coil

coil inductance



Basic Mindset for Fault analysis

Initial analyses :

- We only consider magnetic field as the energy source / torque creation mechanism
- Neglect :
  - Conversion of inertia to current
  - Response of prime mover.

In the remainder of the lecture I just want to sketch this type of analysis for the special case of a 3ϕ fault

"Law of Constant flux"

Note: MBB is sloppy here.

Two principles

1. Conservation of energy
2. Current can't change discontinuously, in an inductor

$\Rightarrow$ MBB use this to justify a law of constant flux: If energy is not dissipated, then flux will remain constant.

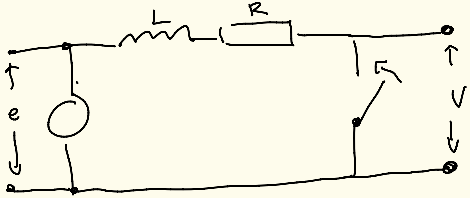
First AC electromagnetic example - single phase.

In the first AC example (Fig 4.2), MBB use just a basic circuit equation:

$$e = E_m \sin(\omega t + \theta_0) = L\frac{di}{dt} + Ri \quad \Rightarrow \quad i(t) = \frac{E_m}{Z}\sin(\omega t + \theta_0 - \phi) - E_m \sin(\theta_0 - \phi)\, e^{-Rt/L}$$

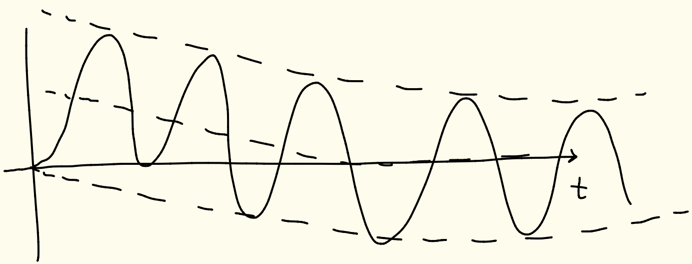
Z: impedance magnitude

θ_0: time of fault



(they don't directly invoke the "law of constant flux")

$\Rightarrow$



$\theta_0 - \phi = -\frac{\pi}{2}$

Note: Initial current is always zero!

3ϕ fault example : What is the fault current?

Total flux linkage, before

$$\psi_A(t) = \psi_{fa} \cos \omega t \quad , \quad \psi_B = \psi_{fa} \cos(\omega t - 2\pi/3) \quad , \quad \psi_C = \psi_{fa} \cos\left(\omega t - \frac{4\pi}{3}\right)$$

$\uparrow$

Flux linkage thru rotor windings

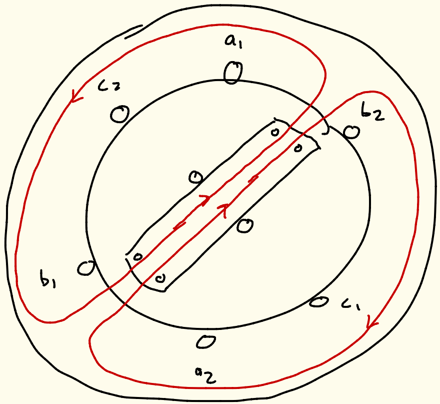
Note, conservation of flux linkage may seem confusing here!

→ That's b/c Flux is changing ctsly before the fault, in each armature winding -

- But the total flux linkage is not.

What MBB seem to be doing is: after the fault, they assume flux linkage in each armature winding is conserved (in the absence of resistive losses)

Thinking Geometrically



— Φ_f : Flux in core (not linkage in an individual winding)

Getting the fault current.

let $\gamma_0 = \omega t$ at time of fault

$$\Rightarrow \Psi_A(t) = \underline{\text{constant}} = \Psi_{AA} + \Psi_{fA} = \Psi_{fa}\cos(\gamma_0)$$

initial flux at time of fault.

Ψ_{AA}: flux originating from armature winding current

Ψ_{fA}: flux linkage from rotor.

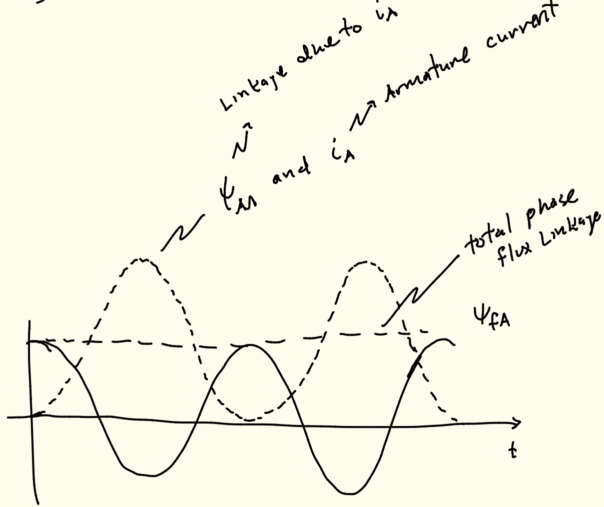
$\Rightarrow$ ① Solve for Ψ_{AA} (field due to fault current)

② Use $i_A = \dfrac{\Psi_{AA}}{L_A}$

i_A: armature current

L_A: Phase winding inductance (dominated by air gap)

$$\Rightarrow i_A = i_m\left(\cos\gamma_0 - \cos(\omega t + \gamma_0)\right)$$



Thinking about the damper

See power point.