Modeling and Simulation Overview

D. M. Auslander

What is a (Digital) Simulation?

- Software that can predict on a moment-by-moment basis the behavior of a system in response to external inputs and disturbances
  - Reference: Computer Control Notes (DigitalControl.pdf or FeedbackBasics.pdf)
- The software is based on a numerical model
- In our case, the numerical model is based on a differential equation

Why?

- Design
  - The system hasn't been built yet but its performance must be predicted
  - The system has been built, but is too expensive or dangerous to experiment with
- Analysis
  - The system is misbehaving in mysterious ways

Example: Basic Motor Model

- Ω is angular velocity
- t is time
- i is applied current
- J is rotary inertia
- k_τ is torque constant
- θ is angular position

$$\frac{d\Omega}{dt} = k_\tau \frac{i}{J}$$

$$\frac{d\theta}{dt} = \Omega$$

Getting the Differential Equation Model

- Engineering ingenuity and creativity!
- Decide on study goals
  - Key step – everything follows from this
- What parts of real system need be considered?
  - Color of motor?
- Mathematical representation – real world is messy!

Numerical Model

- Produces stepwise solution of mathematical model
- For ordinary differential equation (ODE), integrate the right hand side over a small time step
- Dilemma: dependent variables appear on both sides
- Many approximation methods available

Euler's Method Applied to Motor

- Assume dependent variables do not change over step
- Integrand becomes constant
- Easy to implement, not very accurate or stable
- Requires small step size

$$\Omega(t+\Delta t) = \Omega(t) + \frac{k_\tau i}{J}\Delta t$$
$$\theta(t+\Delta t) = \theta(t) + \Omega\Delta t$$

Validation

- Two steps
  - Does the numerical solution correspond to the ODE it comes from?
  - Is it an accurate prediction of the physical system's behavior?
- Important process, difficult and expensive
- Often shortchanged!

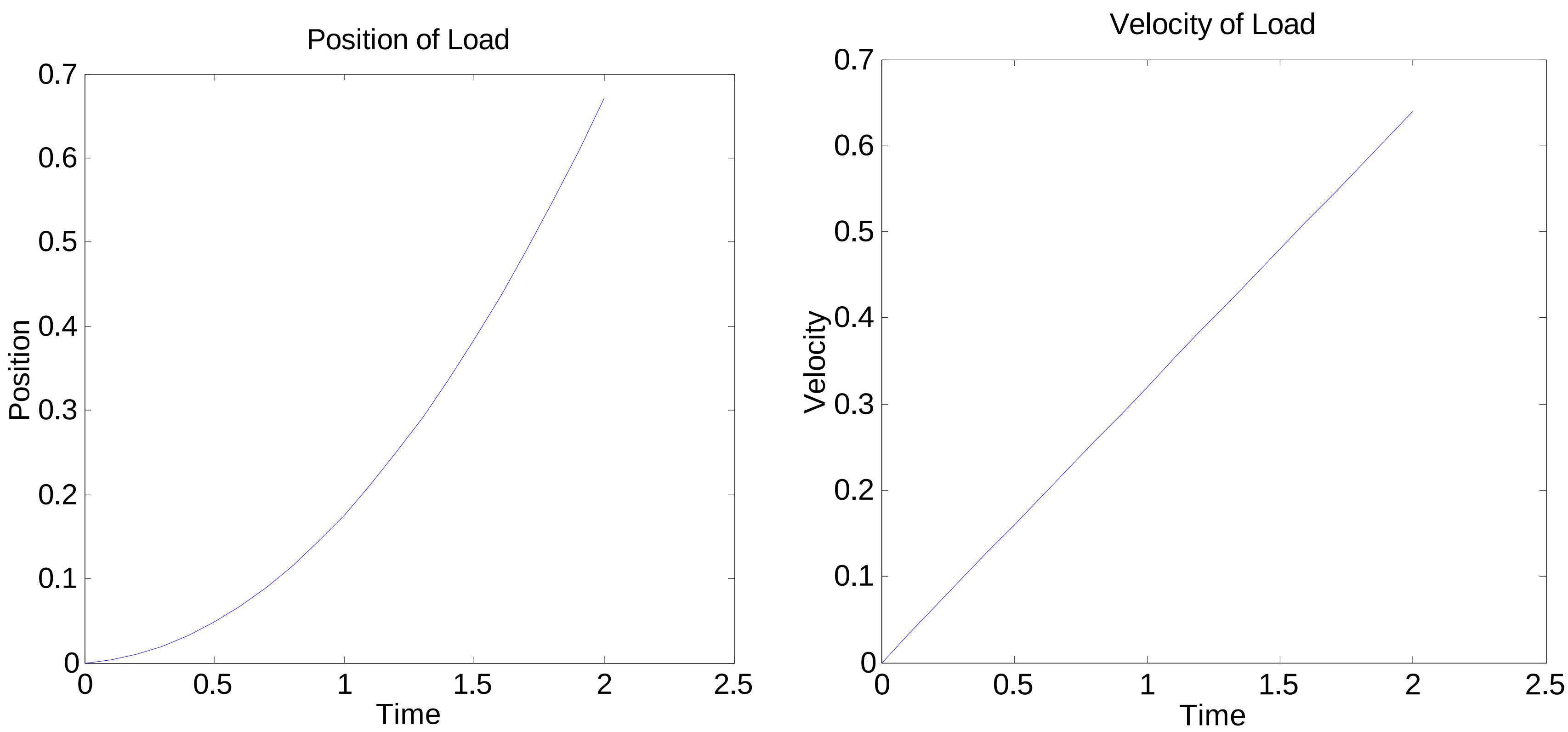
Are the Numerics Right?

- Step size sensitivity
  - If solution is sensitive to step size, step size is too big
  - Step size must be small enough for fastest mode
- Test cases
  - Simplifications for which analytic solutions are available
  - Order-of-magnitude analysis

Is the Prediction Right?

- Much harder to answer
- Does the system exist? Is it testable?
- If yes, experiments will help validation
  - Experiments are expensive
- If no, ???
  - Experience, common sense, luck

Motor Simulation Results (constant applied current)



Improving On Euler's Method

- Standard equation format
- X is state vector
- Equivalent to n-th order form
- Converting from n-th order to this form is easy

…

$$\underline{x} = [x_1, x_2, ..., x_n]$$

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

$$or$$

$$x_1 = f_1(x_1, x_2, ..., x_n, t)$$

$$x_2 = f_1(x_1, x_2, ..., x_n, t)$$

$$...$$

Converting From N-th Order Form

- Define state variables
- This definition is not unique – there are an infinite number of possibilities!
- This choice is very convenient

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 y = u(t)$$

$$define: x_1 = y;\ x_2 = \frac{dy}{dt};\ \ldots;\ x_n = \frac{d^{n-1} y}{dt^{n-1}}$$

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = x_3$$

$$\ldots$$

$$\frac{dx_n}{dt} = \frac{1}{a_n}(-a_{n-1} x_n - \ldots - a_1 x_2 - a_0 x_1)$$

The Formal Solution For One Time Step (Initial Condition Problems)

$$\underline{x}(t+\Delta t) = \underline{x}(t) + \int_{t}^{t+\Delta t} \underline{f}(\underline{x}(\tau), \tau)\, d\tau$$

- The problem is that x is known at time=t (the beginning of the integration period) but not beyond that, so the integrand is unknown!
- ODE solvers are means of approximating the value of this integrand

Runge-Kutta Methods

- Euler method (above) says: approximate f() throughout the integration period with its value at the beginning of the period
- This is very crude!
- It is sometimes OK, but we can do better

Try Going Halfway ...

- Take half a step
- Evaluate
- Use that value to compute f at the middle of the step
- Use the value of f at mid-step to approximate the integrand

$$k_1 = \Delta t\, f(x(t), t)$$

$$k_2 = \Delta t\, f(x(t) + \frac{k_1}{2}, t + \frac{\Delta t}{2})$$

$$x(t + \Delta t) = x(t) + k_2$$

N-Steps → Runge-Kutta

- Break this down to many steps
- Next slide shows formula for 4-th order Runge-Kutta
- Most popular of all integration methods
- Good balance of accuracy, stability and per-step computational cost

4-th Order Runge-Kutta

$$k_1 = \Delta t \ f(x(t), t)$$

$$k_2 = \Delta t \ f(x(t) + \frac{k_1}{2}, t + \frac{\Delta t}{2})$$

$$k_3 = \Delta t \ f(x(t) + \frac{k_2}{2}, t + \frac{\Delta t}{2})$$

$$k_4 = \Delta t \ f(x(t) + k_3, t + \Delta t)$$

$$x(t + \Delta t) = x(t) + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

Use of Runge-Kutta Methods

- These are "explicit" solvers
- Can be stability limited or accuracy limited
- Stability: step size is determined by fastest component of the system, whether it is important for accuracy or not
- Have trouble with "stiff" systems – high ratio of fastest to slowest components

Advanced Topics

- Error Analysis
- Variable step size algorithms
- Implicit methods