Goals:
- Provide a more detailed description of the use of ODEs for modeling
- Provide examples of the type of analysis that can be done using ODEs

Reading:
- Åström and Murray, *Analysis and Design of Feedback Systems*, Ch 2
- Advanced: Lewis, *A Mathematical Approach to Classical Control*, Ch 1
Review: Second Order Differential Equations (Ma 1)

Damped oscillator dynamics

\[ m \ddot{q} + c \dot{q} + kq = f(t) \]

Homogeneous solution: \( f(t) = 0 \)

- Guess form of the solution: \( q(t) = e^{\alpha t} (A \cos \omega t + B \sin \omega t) \)
- Substitute into ODE and solve for the constants

\[
0 = e^{\alpha t} \left( \left( B(c + 2\alpha m) \omega + A \left( m\alpha^2 + c\alpha - m\omega^2 + k \right) \right) \cos(\omega t) \\
+ \left( Bm\alpha^2 + Bc\alpha - 2Am\omega\alpha - Bm\omega^2 + Bk - Ac\omega \right) \sin(\omega t) \right)
\]

\[
q_0 = A \\
v_0 = A\alpha + B\omega
\]

- Solve for \( A \) & \( B \)

- Coefficients of \( \sin/\cos \) must be zero
- Use to solve for \( \alpha, \omega \)

- Simplify the solution by pulling out common terms

\[
q(t) = e^{-\zeta \omega_0 t} \left( q_0 \cos \omega_d t + \left( \frac{\zeta \omega_0}{\omega_d} q_0 + \frac{1}{\omega_d} v_0 \right) \sin \omega_d t \right)
\]

- Note: this solution holds when \( \zeta < 1 \)
Second Order Differential Equations, ctd

\[ m\ddot{q} + c\dot{q} + kq = f(t) \]

**Particular response: zero initial conditions**
- \( q(0) = 0, \dot{q}(0) = 0 \)
- Response to constant (step) input, \( f(t) = F \)

\[ q(t) = \frac{F}{m\omega_0^2} \left( 1 - e^{-\zeta\omega_0 t} \cos \omega_d t + \frac{\zeta}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_0 t} \sin \omega_d t \right) \]

- Response to sinusoidal input, \( f(t) = A \sin \omega t \)

\[ q(t) = MA \sin(\omega t + \theta) \quad Me^{j\theta} = \frac{\omega^2}{\omega_0^2 - \omega^2 + 2j\zeta\omega_0\omega}. \]

- Form of the solution: sinusoid at same frequency, with shift in mag & phase
- Solving by hand is a mess; we will learn much better ways later

**Complete solution: homogeneous + particular**
More General Forms of Differential Equations

State space form

\[
\frac{dx}{dt} = f(x, u) \quad \quad y = h(x, u)
\]

General form

\[
\frac{dx}{dt} = Ax + Bu \quad \quad y = Cx + Du
\]

Linear system

\[
x \in \mathbb{R}^n, \; u \in \mathbb{R}^p \quad \quad y \in \mathbb{R}^q
\]

Higher order, linear ODE

\[
\frac{d^{n+q}x}{dt^{n+q}} + a_1\frac{d^{n+q-1}x}{dt^{n+q-1}} + \cdots + a_nx = u
\]

\[
y = b_1\frac{d^{n+q-1}x}{dt^{n+q-1}} + \cdots + b_{n-1}\dot{x} + b_nx
\]

\[
x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{d^{n-1}x}{dt^{n-1}} \\ \vdots \\ \frac{dx}{dt} \\ x \end{bmatrix}
\]

\[
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u
\]

\[
y = [b_1 \ b_2 \ \cdots \ b_n] x + du.
\]
Analytical Solutions of ODEs

Scalar systems

\[
\frac{dx}{dt} = ax + u \\
y = x
\]

\[
x_h(t) = e^{at}x_0 \\
u = A \sin \omega_1 t \\
y = -A \frac{-\omega_1 e^{at} + \omega_1 \cos \omega_1 t + a \sin \omega_1 t}{a^2 + \omega_1^2}
\]

Decoupled systems

\[
\frac{dx}{dt} = \begin{bmatrix} \lambda_1 & 0 & \cdots \\ 0 & \lambda_2 & \cdots \\ \vdots & \ddots & \ddots \end{bmatrix} x + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} u
\]

\[
y = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \end{bmatrix} x + Du.
\]

\[
\dot{x}_i = \lambda_i x_i + \beta_i u \\
x_i(t) = e^{\lambda_i t} x(0) + \int_0^t e^{\lambda_i (t-\tau)} \beta_i u(\tau) \, d\tau.
\]

- Effect of input modeled by “convolution integral”

General solutions

- Linear systems: use Jordan canonical form and “matrix exponential” (more later)
- Nonlinear system: generally no closed form solutions, expect in special cases
Numerical Solution of ODEs

Numerical simulation: Euler integration

\[ \frac{dx}{dt} = \lim_{\epsilon \to 0} \frac{x(t + \epsilon) - x(t)}{\epsilon} \quad \Longrightarrow \quad x(t + \epsilon) \approx x(t) + \epsilon f(x(t), u(t)) \]

- If \( \epsilon \) chosen sufficiently small, get good approximation analytical solution
- Solution is in the form of a difference equation (with step size \( \epsilon \))

More accurate algorithms: build better approximation to the derivative
- Faster algorithms: choose the step size based on how quickly solution is changing
- Example: Runga Kutta (ode45)
Analyzing Models using ODEs: Frequency Response

How does linear system respond to sinusoidal inputs?

\[ m \ddot{q} + c \dot{q} + kq = f(t). \]
\[ f(t) = A \sin \omega t. \]
\[ q(t) = g(\omega) \sin(\omega t + \phi(\omega)), \]

General properties

- Linear systems: sinusoidal input at frequency \( \omega \) \( \Rightarrow \) sinusoidal output at frequency \( \omega \)
- Gain: \( \frac{\text{output magnitude}}{\text{input magnitude}} = \frac{g(\omega)}{A} \)
- Phase: shift in input sinusoid versus output sinusoid

CDS 110, 4 Oct 06  R. Murray/H. Mabuchi, Caltech
# Analyzing Models Using ODEs: Stability

ODEs can also be used to prove stability of a system:

- Try to reason about the long term behavior of *all* solutions
- Stability: all solutions return to equilibrium point (more precise defn later)
  \[
  m\ddot{q} + c\dot{q} + kq = 0
  \]

### Example: spring mass system

- Can we show that all solutions return to rest w/out explicitly solving ODE?
- Idea: look at how energy evolves in time

- Start by writing equations in state space form
- Compute energy and its derivative

\[
V(x) = \frac{1}{2}kx_1^2 + \frac{1}{2}mx_2^2, \quad \frac{dV}{dt} = kx_1\dot{x}_1 + mx_2\dot{x}_2 = kx_1x_2 + mx_2(-\frac{b}{m}x_2 - \frac{k}{m}x_1) = -bx_2^2,
\]

- Energy is positive \(\Rightarrow x_2\) must eventually go to zero
- If \(x_2\) goes to zero, can show that \(x_1\) must also approach zero (Lasalle, W3)
Modeling from Experiments

Example: spring mass system
• Measure response of system to a step input

\[ q(t) = \frac{F_0}{k} \left( 1 - e^{-\frac{bt}{2m}} \left[ \cos\left(\frac{\sqrt{4km-b^2}}{2m} t\right) - \frac{1}{\sqrt{4km-b^2}} \sin\left(\frac{\sqrt{4km-b^2}}{2m} t\right) \right] \right) \]

\[ \frac{2\pi}{T} = \frac{\sqrt{4km-b^2}}{2m} \]

\[ \log(q(t_1) - \frac{F_0}{k}) - \log(q(t_2) - \frac{F_0}{k}) = \frac{b}{2m} (t_2 - t_1) \]

\[ q(\infty) = \frac{F_0}{k} \]
Block Diagrams

Block diagrams separate components of a system into manageable units

Example: cruise control
• Each block corresponds to a portion of the overall dynamics
• Write out the individual blocks as input/output systems

Body
• Dynamics: \( m \frac{dv}{dt} = F - F_d \).
• State: \( v \) - velocity of vehicle
• Inputs: \( F, F_d \) - force from wheels, external disturbances (wind, hills, etc)
• Output: \( v \) - velocity of vehicle

Dynamic versus state blocks
• Some blocks represent static relationships (no states); eg, gears and wheels
Standard Block Diagram Notation

Remarks
• SIMULINK uses slightly different symbols in a few places (eg, gain block)
Example: Hovering Mesoscale Robot (HOMER)

Project Goals

- Characterize and reverse engineer the sensory-motor control system of the fly
- Apply salient features to the design of micro air vehicles and other autonomous systems
- Experimentation and modeling key components of flight control system: (1) take-off, (2) robustness to wing gust, (3) chemical tracking, and (4) sensory fusion (visual, gyro)
Vision as a Compensatory Mechanism for Disturbance Rejection in Upwind Flight

Michael Reiser    Sean Humbert
Domitilla Del Vecchio    Mary Dunlop
Michael Dickinson    Richard Murray

Project results
• Interconnected simplified models that provide bio-realistic behavior for upwind flight

Insights
• Low level (fast!) vision and sensory motor processing capable of generated complex behaviors that achieve desired response
Vision-Based Navigation Using Wide-Field Integration
Sean Humbert (U. Maryland)

• Approach
  • Understand & characterize wide field integration processing in Drosophila
  • Near 360° optical flow processing
  • Very fast coupling to flight actuation

Flight Stabilization and Obstacle Avoidance
Engineering Applications in Vision-Based Navigation
Preview: Linear Control Systems and Convolution

\[
\dot{x} = Ax + Bu \\
y = Cx + Du
\]

Impulse response, \( h(t) = Ce^{At}B \)
- Response to input “impulse”
- Equivalent to “Green’s function”

Linearity \( \Rightarrow \) compose response to arbitrary \( u(t) \) using convolution
- Decompose input into “sum” of shifted impulse functions
- Compute impulse response for each
- “Sum” impulse response to find \( y(t) \)

Complete solution: use integral instead of “sum”
\[
y(t) = Ce^{At}x(0) + \int_{\tau=0}^{t} Ce^{A(t-\tau)} Bu(\tau) d\tau + Du(t)
\]
- linear with respect to initial condition and input
- 2X input \( \Rightarrow \) 2X output when \( x(0) = 0 \)