CDS 101/110a: Lecture 1.2
System Modeling

Richard M. Murray
1 October 2008

Goals:
• Define a “model” and its use in answering questions about a system
• Introduce the concepts of state, dynamics, inputs and outputs
• Review modeling using ordinary differential equations (ODEs)

Reading:
• Åström and Murray, Feedback Systems, Sections 2.1–2.3, 3.1 [40 min]
• Advanced: Lewis, A Mathematical Approach to Classical Control, Chapter 1
Model-Based Analysis of Feedback Systems

Analysis and design based on models

- A model provides a *prediction* of how the system will behave
- Feedback can give counter-intuitive behavior; models help sort out what is going on
- For control design, models don’t have to be exact: *feedback* provides robustness

Control-oriented models: inputs and outputs

The model you use depends on the questions you want to answer

- A single system may have many models
- Time and spatial scale must be chosen to suit the questions you want to answer
- Formulate questions *before* building a model

Weather Forecasting

- Question 1: how much will it rain tomorrow?
- Question 2: will it rain in the next 5-10 days?
- Question 3: will we have a drought next summer?

Different questions ⇒ different models
Example #1: Spring Mass System

Applications
- Flexible structures (many apps)
- Suspension systems (eg, “Bob”)
- Molecular and quantum dynamics

Questions we want to answer
- How much do masses move as a function of the forcing frequency?
- What happens if I change the values of the masses?
- Will Bob fly into the air if I take that speed bump at 25 mph?

Modeling assumptions
- Mass, spring, and damper constants are fixed and known
- Springs satisfy Hooke’s law
- Damper is (linear) viscous force, proportional to velocity
Modeling a Spring Mass System

Model: rigid body physics (Ph 1)
- Sum of forces = mass * acceleration
- Hooke’s law: \( F = k(x - x_{\text{rest}}) \)
- Viscous friction: \( F = c \, v \)

Converting models to state space form
- Construct a vector of the variables that are required to specify the evolution of the system
- Write dynamics as a system of first order differential equations:

\[
\begin{align*}
\frac{dx}{dt} &= f(x, u) \quad x \in \mathbb{R}^n, \; u \in \mathbb{R}^p \\
y &= h(x) \quad y \in \mathbb{R}^q
\end{align*}
\]

\[
\begin{bmatrix}
\dot{q}_1 \\
\dot{q}_2
\end{bmatrix}
= 
\begin{bmatrix}
\frac{k_2}{m} (q_2 - q_1) - \frac{k_1}{m} q_1 \\
\frac{k_3}{m} (u - q_2) - \frac{k_2}{m} (q_2 - q_1) - \frac{c}{m} \dot{q}
\end{bmatrix}
\]

\[
y = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}
\]

“State space form”
**Review: Second Order Differential Equations**

\[ m\ddot{q} + c\dot{q} + kq = u \]

**Particular response: zero initial conditions**

- \( q(0) = 0, \ \dot{q}(0) = 0 \)
- Response to constant (step) input, \( u(t) = F \)

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

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

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

- 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**

- Warning: be careful to make sure the initial conditions are satisfied
More General Forms of Differential Equations

State space form
\[
\begin{align*}
\frac{dx}{dt} &= f(x, u) \\
y &= h(x, u)
\end{align*}
\]

General form
Linear system
\[
\begin{align*}
\frac{dx}{dt} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]
\[x \in \mathbb{R}^n, u \in \mathbb{R}^p\]
\[y \in \mathbb{R}^q\]

Higher order, linear ODE
\[
\begin{align*}
d^nq &= a_1\frac{d^{n-1}q}{dt^{n-1}} + \cdots + a_nq = u \\
y &= b_1\frac{d^{n-1}q}{dt^{n-1}} + \cdots + b_{n-1}q + b_nq
\end{align*}
\]
\[
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{n-1} \\
x_n
\end{bmatrix} =
\begin{bmatrix}
\frac{d^{n-1}q}{dt^{n-1}} \\
\frac{d^{n-2}q}{dt^{n-2}} \\
\vdots \\
\frac{dq}{dt} \\
q
\end{bmatrix}
\]
\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_n
\end{bmatrix} =
\begin{bmatrix}
-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
er
\end{bmatrix}
\]
\[
y = \begin{bmatrix} \phantom{-}b_1 \\
\phantom{-}b_2 \\
\vdots \\
\phantom{-}b_n
\end{bmatrix} x
\]
Simulation of a Mass Spring System

Steady state frequency response
- Force the system with a sinusoid
- Plot the “steady state” response, after transients have died out
- Plot relative magnitude and phase of output versus input (more later)

Matlab simulation (see handout)
function dydt = f(t, y, ...)
u = 0.00315*cos(omega*t);
dydt = [
y(3);
y(4);
-(k1+k2)/m1*y(1) + k2/m1*y(2);
k2/m2*y(1) - (k2+k3)/m2*y(2) - c/m2*y(4) + k3/m2*u ];

[t,y] = ode45(dydt,tspan,y0,[],k1,k2,k3,m1,m2,c,omega);
Modeling Terminology

**State** captures effects of the past
- independent physical quantities that determines future evolution (absent external excitation)

**Inputs** describe external excitation
- Inputs are *extrinsic* to the system dynamics (externally specified)

**Dynamics** describes state evolution
- update rule for system state
- function of current state and any external inputs

**Outputs** describe measured quantities
- Outputs are function of state and inputs \( \Rightarrow \) not independent variables
- Outputs are often *subset* of state

**Example: spring mass system**
- State: position and velocities of each mass: \( q_1, q_2, \dot{q}_1, \dot{q}_2 \)
- Input: position of spring at right end of chain: \( u(t) \)
- Dynamics: basic mechanics
- Output: measured positions of the masses: \( q_1, q_2 \)
Modeling Properties

Choice of state is not unique
- There may be many choices of variables that can act as the state
- Trivial example: different choices of units (scaling factor)
- Less trivial example: sums and differences of the mass positions

Choice of inputs and outputs depends on point of view
- Inputs: what factors are external to the model that you are building
  - Inputs in one model might be outputs of another model (e.g., the output of a cruise controller provides the input to the vehicle model)
- Outputs: what physical variables (often states) can you measure
  - Choice of outputs depends on what you can sense and what parts of the component model interact with other component models

Can also have different types of models
- Ordinary differential equations for rigid body mechanics
- Finite state machines for manufacturing, Internet, information flow
- Partial differential equations for fluid flow, solid mechanics, etc
Difference Equations

Difference equations model discrete transitions between continuous variables

- “Discrete time” description (clocked transitions)
- New state is function of current state + inputs
- State is represented as a continuous variable

Example: predator prey dynamics

Questions we want to answer

- Given the current population of hares and lynxes, what will it be next year?
- If we hunt down lots of lynx in a given year, how will the populations be affected?
- How do long term changes in the amount of food available affect the populations?

Modeling assumptions

- Track population annual (discrete time)
- The predator species is totally dependent on the prey species as its only food supply
- The prey species has an external food supply and no threat to its growth other than the specific predator.

\[
x[k + 1] = f(x[k], u[k])
\]
\[
y[k] = h(x[k])
\]
Example #2: Predator Prey Modeling

Discrete Lotka-Volterra model

- **State**
  - $H[k]$  
    - # of rabbits in period $k$
  - $L[k]$  
    - # of foxes in period $k$

- **Inputs (optional)**
  - $u[k]$  
    - amount of rabbit food

- **Outputs:**  
  - # of rabbits and foxes

- **Dynamics:** Lotka-Volterra eqs

\[
H[k+1] = H[k] + b_r(u)H[k] - aL[k]H[k],
\]
\[
L[k+1] = L[k] + cL[k]H[k] - d_fL[k],
\]

- **Parameters/functions**
  - $b_r(u)$  
    - hare birth rate (per period);  
      depends on food supply
  - $d_f$  
    - lynx mortality rate (per period)
  - $a, c$  
    - interaction terms

MATLAB simulation (see handout)

- Discrete time model, “simulated” through repeated addition

Comparison with data
Summary: System Modeling

Model = state, inputs, outputs, dynamics

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

**Principle:** Choice of model depends on the questions you want to answer

```
function dydt = f(t,y, k1, k2, k3, m1, m2, c, omega)
    u = 0.00315*cos(omega*t);
    dydt = [y(3);
            y(4);
            -(k1+k2)/m1*y(1) + k2/m1*y(2);
            k2/m2*y(1) - (k2+k3)/m2*y(2) - b/m2*y(4) + k3/m2*u ];
```

\[
x[k + 1] = f(x[k], u[k]) \\
y[k] = h(x[k])
\]