CDS 101: Lecture 2.1
System Modeling

Richard M. Murray
7 October 2002

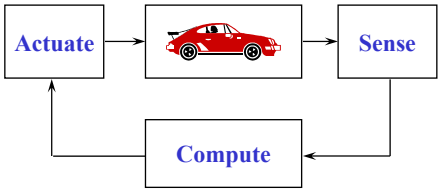
Goals:

- Describe what a model is and what types of questions it can be used to answer
- Introduce the concepts of state, dynamic, and inputs
- Provide examples of common modeling techniques: finite state automata, difference equations, differential equations, Markov chains
- Describe common modeling tradeoffs

Reading:

- K. J. Astrom, *Control Systems Design*, Sections 3.1-3.2, 3.6
- *Optional*: Astrom, Section 3.3

Review from last week



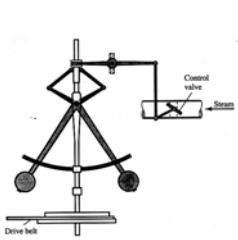
Control =

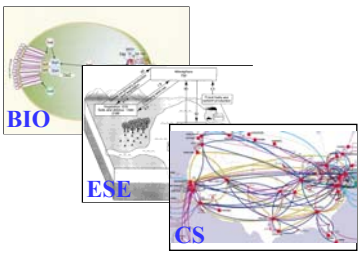
Sensing + Computation + Actuation

Feedback Principles

- Robustness to Uncertainty
- Design of Dynamics

Many examples of control and feedback in natural and engineered systems:





30 Sep 02 R. M. Murray, Caltech CDS 0

Models

Models are a mathematical *representations* of system dynamics

- Models allow the dynamics to be simulated and analyzed, without having to build the system
- Models are *never* exact, but they can be *predictive*

Models can be used in ways that the system can't

- Certain types of analysis (eg, parametric variations) can't easily be done on the actual system
- In many cases, models can be run much more quickly than the original models

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
- *Always* formulate questions *before* building a model

Example: 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

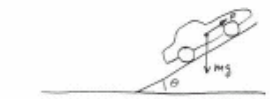
7 Oct 02 R. M. Murray, Caltech CDS 3

Physical Concept of State

A key concept in modeling is the concept of *state*

The *state* of a model of a dynamic system is a set of independent physical quantities, the specification of which (in the absence of excitation) completely determines the future evolution of the system

Example #1: car on a sloping road



- State: position and velocity of car
- Angle of incline is *not* a state (not part of the model of the car)
- Accelerator position is *not* a state (not *intrinsic* to the car)

Example #2: predator prey (rabbits vs foxes)

- State: number of rabbits and foxes
- Amount of rabbit food is *not* a state (not intrinsic to the ecosystem as we have defined it)
- Number of dead rabbits is *not* a state (not independent of number of live rabbits)

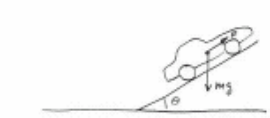
7 Oct 02 R. M. Murray, Caltech CDS 4

Dynamics

***Dynamics* describes how the state evolves**

The *dynamics* of a model is an update rule for the system state that describes how the state evolves, as a function on the current state and any external inputs

Example #1: car on a sloping road



- Dynamics: Newton's law (F = ma)

$$m\ddot{x} = -b\dot{x} + u_{\text{engine}}(t) + u_{\text{hill}}(t)$$

- Engine force modeled as external input
- Hill modeled as external input

Example #2: predator prey

- Dynamics: empirically observed difference eqs

$$R[k+1] = R[k] + b_r R[k] - aR[k]F[k]$$
$$F[k+1] = F[k] - d_f F[k] + aR[k]F[k]$$

- *System* of difference equations

7 Oct 02 R. M. Murray, Caltech CDS 5

Inputs

Inputs describe the *external excitation* of the dynamics

- Inputs are *extrinsic* to the system dynamics (externally specified)
- Constant inputs are often considered to be *parameters*

Example #1: car on a sloping road

$$m\ddot{x} = -b\dot{x} + u_{\text{engine}}(t) + u_{\text{hill}}(t)$$

Input #1 Input #2

Example #2: predator prey

$$R[k+1] = R[k] + \mathbf{b_r(u)}R[k] - aR[k]F[k]$$
$$F[k+1] = F[k] - d_f F[k] + aR[k]F[k]$$

- Rabbit food can either be a parameter (if constant) or an external input (if nonconstant)

7 Oct 02 R. M. Murray, Caltech CDS 6

Outputs

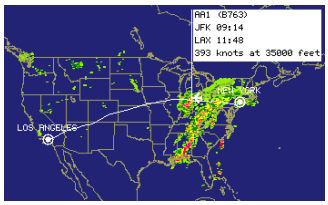
Outputs describe the directly measured variables

- Outputs are a function of the state and inputs ⇒ not independent variables
- Not all states are outputs; some states can't be directly measured

Example #1: car on a sloping road

- Outputs: position and velocity
- Measure velocity with speedometer
- Measure position with odometer (or GPS)

Example #2: motion of an airplane over US



http://www.flightview.com/

- States: position, altitude, linear + angular vel
- Dynamics: aerodynamics of flight
- Inputs: thrust, rudder, elevator, flaps, wind
- Outputs (from radar): heading and speed
- Roll, pitch and yaw are not *directly* measurable; part of state, but not part of outputs

Modeling Properties

Choice of state is not unique

- There may be *many* choices of variables that can act as the state
- Predator prey example: look at number of rabbits plus the *excess* of foxes over rabbits:

$$\begin{aligned} R[k+1] &= f_r(R[k], F[k]) \\ F[k+1] &= f_e(R[k], F[k]) \end{aligned} \xrightarrow{E = F - R} \begin{aligned} R[k+1] &= f_r(R[k], E[k]) \\ E[k+1] &= f_e(R[k], E[k]) \end{aligned}$$

- Can also look at models for the same system with *different* numbers of states
  - Ignore certain physical effects (and hence eliminate states)

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 (eg, 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

Model Types

Models are a mathematical *representations* of system dynamics

- Models allow the dynamics to be simulated and analyzed, without having to build the system
- Models are *never* exact, but they can be *predictive*

Different types of models are used for different purposes

- Ordinary differential equations for rigid body mechanics
- Finite state machines for manufacturing, Internet, information flow
- Partial differential differential equations for fluid flow, solid mechanics, etc

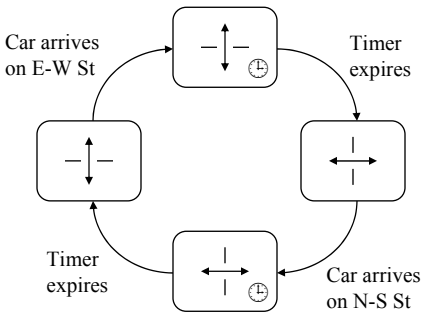
7 Oct 02 R. M. Murray, Caltech CDS 9

Model Type #1: Finite State Machines

Finite state machines model discrete transitions between finite number of states

- Represent each configuration of system as a state
- Model transition between states using a graph
- Inputs force transition between states

Example: Traffic light logic



State:	current pattern of lights that are on + internal timers
Inputs:	presence of car at intersections
Outputs:	current pattern of lights that are on

7 Oct 02 R. M. Murray, Caltech CDS 10

Model Type #2: Difference Equations

Difference equations model discrete transitions between state variables

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

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

Example: predator prey

$$R[k+1] = R[k] + \boldsymbol{b_r(u)}R[k] - aR[k]F[k]$$
$$F[k+1] = F[k] - d_f F[k] + aR[k]F[k]$$

- Note that state is continuous $\Rightarrow$ models average number of rabbits, foxes

State: number of rabbits and foxes
Inputs: rabbit food available
Outputs: number of rabbits and foxes

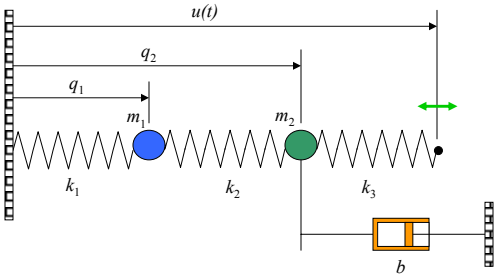
Model Type #3: Differential Equations

Differential equations model continuous evolution of state variables

- Describe the rate of change of the state variables
- Both state and time are continuous variables

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

Example: Coupled spring mass system



$$m_1\ddot{q}_1 = k_2(q_2 - q_1) - k_1 q_1$$
$$m_2\ddot{q}_2 = k_3(u - q_2) - k_2(q_2 - q_1) - b\dot{q}_2$$

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \frac{k_2}{m}(q_2 - q_1) - \frac{k_1}{m}q_1 \\ \frac{k_3}{m}(u - q_2) - \frac{k_2}{m}(q_2 - q_1) - \frac{b}{m}\dot{q}_2 \end{bmatrix}$$

$$y = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

"State space form"

State: positions and velocities of masses
Inputs: motion of end of rightmost spring
Outputs: positions of masses (via sensors)

Model type #4: Markov chains

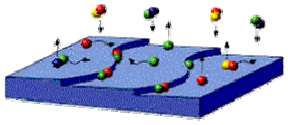
Markov chains model continuous or discrete evolution of probabilities

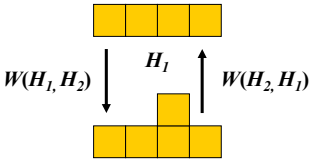
- States correspond to probabilities of events
- Dynamics can be continuous or discrete time
- Very useful for stochastic systems

$$\frac{d}{dt}P_H = \sum_{H'} W_u(H',H)P_{H'} - W_u(H,H')P_H$$

states inputs transition rates

Example: Molecular models for materials growth





Questions

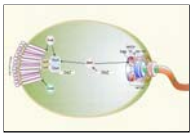
- What is the roughness of the final surface?
- What are the film properties?
- How do we modify the film properties by varying the processing conditions under sensor feedback (control)?

State:	probabilities of being in given states
Inputs:	processing conditions (temp, flow)
Outputs:	surface profile, film properties

Comments on Model Type

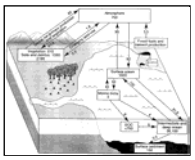
One system may have many choices for type of model

- Mass spring system sampled be a computer could use a continuous time description (what your eyes see) or a discrete time description (what the computer sees)
- *The choice of model type depends on the question you want to answer*



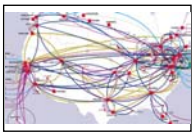
Many other types of models besides those listed

- Partial differential equations (PDEs)
- Hybrid systems: mixed discrete and continuous time/variables



This class will focus primarily on <u>ordinary differential equations (ODEs)</u>

- Wide applicability + lots of tools
- Homework will explore discrete time analogs to continuous time results



Building Models (A True Story)

GM Astro "tractor"

Problem reported from the field

- Truck would start "bucking" when going up hill, in the mountains, in first gear, under full load (trailer)
- GM Delco Systems Operations called on to solve the problem (and they assigned me!)

Model #1: cruise control

- Simplest possible model
- Included simple engine dynamics (1st order ODE) + vehicle properties
- Didn't demonstrate instability

Model #2: add detailed engine dynamics

- Added multi-state, nonlinear model of engine dynamics
- Took into account engine RPM, engine loading, throttle position, etc

Experiment #2: field tests

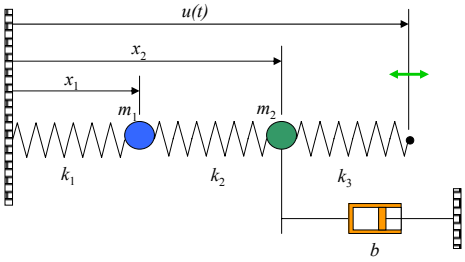
- 1st gear, use brakes to simulate load, *lots* of smoke
- Forced response testing to get system model (ODEs)

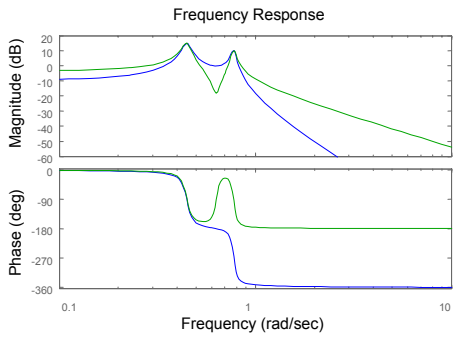
Model #3: transmission model

- Long drive shaft, with loading
- Increased stiffness in shaft under loading recreated the field data (!)

7 Oct 02 R. M. Murray, Caltech CDS 15

Example #1: mass spring system (1/2)





Applications of mass/spring systems

- Flexible structures (buildings, wings, membranes, many more…)
- Molecular dynamics, potential wells

Questions we want to answer

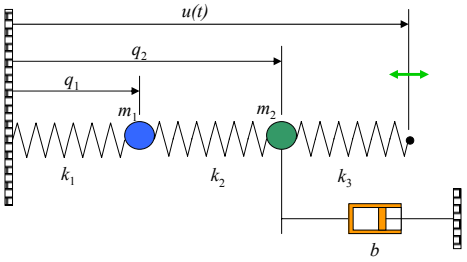
- How much do the masses move as a function of the forcing frequency?
- What happens if I change the values of the masses?
- Will my building stay up in an earthquake?

Modeling assumptions

- Mass, spring, and damper constants are fixed and known
- Springs satisfy Hooke's law
- Damper is (linear) viscous force, proportional to velocity

7 Oct 02 R. M. Murray, Caltech CDS 16

Example #1: mass spring system (2/2)



Rigid Body Physics

- Sum of forces = mass * acceleration
- Hooke's law: $F = k(x - x_{\text{rest}})$
- Viscous friction: $F = b\,v$
- Input: changes end point of spring 3

$$m_1\ddot{q}_1 = k_2(q_2 - q_1) - k_1 q_1$$
$$m_2\ddot{q}_2 = k_3(u - q_2) - k_2(q_2 - q_1) - b\dot{q}_2$$

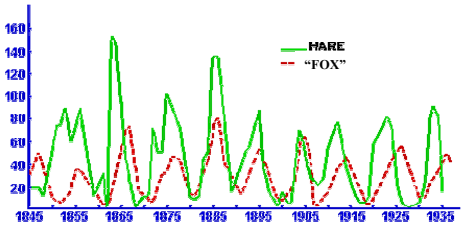
Matlab simulation

```
# twomass.m - CDS 101 example

function dydt = f(t,y, k1, k2,
k3,   m1, m2, b, 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 ];

[t,y] = ode45(dydt,tspan,y0,[],
k1, k2, k3, m1, m2, b, omega);
```


Example #2: Predator Prey (1/2)



http://www.math.duke.edu/education/ccp/materials/diffeq/predprey/contents.html

Questions we want to answer

- Given the current population of rabbits and foxes, what will it be next year?
- If we hunt down lots of foxes in a given year, what will the effect on the rabbit and fox population be?
- How do long term changes in the amount of rabbit food available affect the populations?

Modeling assumptions

- 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.

Example #2: Predator Prey (2/2)

Discrete Lotka-Volterra model

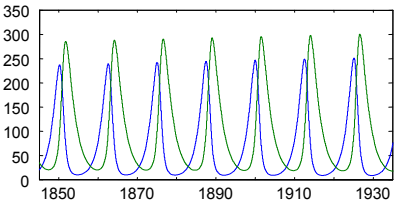
- State
  - $R[k]$ number of rabbits in period k
  - $F[k]$ number of foxes in period k
- Inputs (optional)
  - $u[k]$ amount of rabbit food
- Outputs: number of rabbits and foxes
- Dynamics: discrete Lotka-Volterra eqs

$$R[k+1] = R[k] + b_r(u)R[k] - aF[k]R[k]$$
$$F[k+1] = F[k] - d_f F[k] + aF[k]R[k]$$

- Parameters/functions
  - $b_r(u)$ annual birth rate of rabbits (depends on food supply)
  - d_r annual death rate for foxes
  - a interaction term

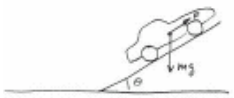
Matlab simulation

```
# predprey.m - CDS 101 example
R[1]=20; F[1]=35;
br = 0.7; df = 0.5; a = 0.007;
nperiods=208; year(1)=1845;
for k=1:90*nperiods
 R(k+1) = R(k) + (br*R(k) -
   a*F(k)*R(k))/nperiods;
 F(k+1) = F(k) + (-df*F(k) +
   a*F(k)*R(k))/nperiods;
 year(k) = year(k) + 1/nperiods;
end;
```

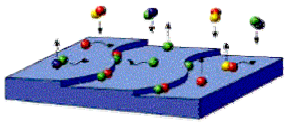


Summary: System Modeling

Model = state, inputs, outputs, dynamics

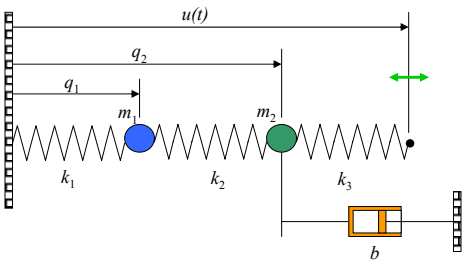


$$\frac{dx}{dt} = f(x,u)$$
$$y = h(x)$$



$$\frac{d}{dt}P_H = \sum_{H'} W_u(H',H)P_{H'} - W_u(H,H')P_H$$

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




```
function dydt = f(t,y, k1, k2,
k3,   m1, m2, b, 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 ];
```