Today: State Space: Matrix Equations

Lecture 9
ECE 147B
Digital Control

Reading: FPW 6.1
or: FV 7.6, 7.7

Topics:
- Controls overview (high-level viewpoint)
- **State Space Equations**
- Numerical approx's: NOW VIA MATRIX EQNS!!
- ZOH effect: NOW VIA MATRIX EQNS!!
- MATLAB commands (in the land of state space…)
- Review for Midterm (to be continued…)
- Midterm evaluations

Next Class: Midterm Review and Midterm Course Evaluations.
- Review all material to date, especially class handouts

Controls Overview: Why "State Space"?

Controls techniques can be grouped, broadly-speaking, into one of two camps:
1. Transform Methods (i.e., using Laplace, Z, and/or Fourier transforms)
2. State Space Methods

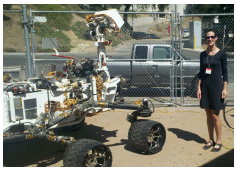
Here, at mid-term in 147B, we cross over from 1 to introduce 2.

But first, an overview in: **Approaches to Control** (…state space vs transforms…)

"Classical Control" (aka **Transform Methods**)	"Modern Control" (aka **State Space Methods**)
Transforms allow one to understand and manipulate dynamics "intuitively".	Matrix equations allow for multi-input multi-output (MIMO) control.
e.g., Root locus, Bode, Nyquist, "Direct Design" in z domain, etc.	e.g., Pole placement, linear quadratic regulator (LQR) control, etc.
Can do calculations BY HAND.	COMPUTERS make this a practical approach, as system complexity goes up.
But normally used for single-input single-output (SISO), rather than MIMO.	(But not as "intuitive"…)

Recall our motivation for learning digital control...

- We want to use computers to control cool stuff in the real world... MIMO control needed.



robot arm

dish washers & cloth washers

stand-alone CNC machine millers

4-legged

6-legged

& more legged robots

automated guided vehicles

unmanned aerial vehicles

Humanoid robots & 2-legged robots

wheeled robots

caterpillar-tracked robots

- ...so, where did Modern Control all begin???

State Space Equations

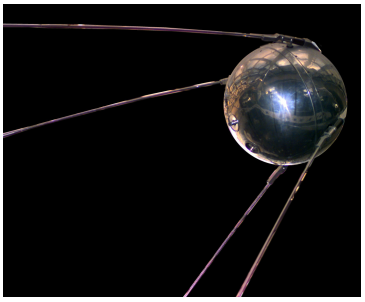
1957: The space race is off! Dawning of the age of "Modern Control" - aka "State Space Methods".

1957 Computers: IBM switches from vacuum tubes to transistors.

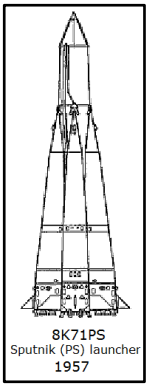
1957 Space Sci.: Sputnik orbits earth!

1957 Rockets: R7 launch vehicle (for Sputnik). Gateway to ICBMs... "Rocket science"

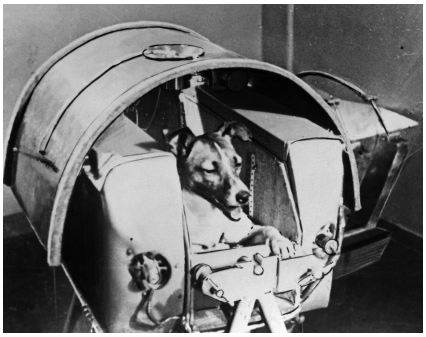
1957 Defense: ARPA (now "DARPA") founded.



Sputnik 1
[replica, Air and Space Museum]



8K71PS
Sputnik (PS) launcher
1957



Sputnik 2 [with stray-dog "hero" Laika]

Matrix form Equations of Motion

CT:

$$\dot{X} = AX + Bu$$
$$y = CX + Du$$

matrix equations for multi-input, multi-output systems ("MIMO")

DT:

$$X(k+1) = A_d X(k) + B_d u(k)$$
$$y(k) = C_d X(k) + D_d u(k)$$

1. "State space" eqns (above) describe LINEAR, time-invariant (LTI) dynamics.
2. An n^{th}-order system has n STATES (in vector X).
3. There will be n "state equations" to represent n^{th}-order dynamics.
4. The number of outputs, m, in output vector Y may be less than, equal to, or greater than the number of states, n, in X.

Matrix form Equations of Motion

CT:

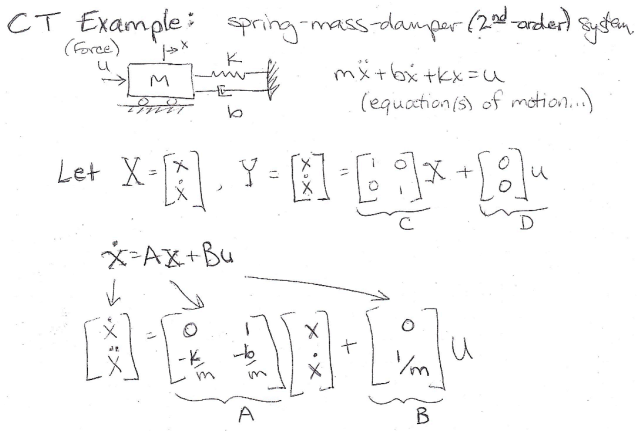
$$\dot{X} = AX + Bu$$
$$y = CX + Du$$

Toy example 1: (1st-order system)

$$\frac{V_c(s)}{V_{in}(s)} = \frac{1}{RCs+1}$$

Matrix form Equations of Motion (cont…)

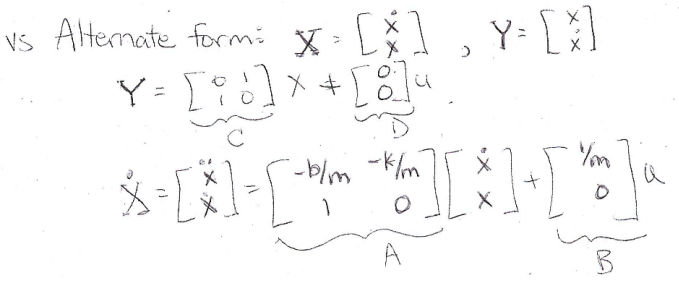
Toy example 2: (2nd-order system)



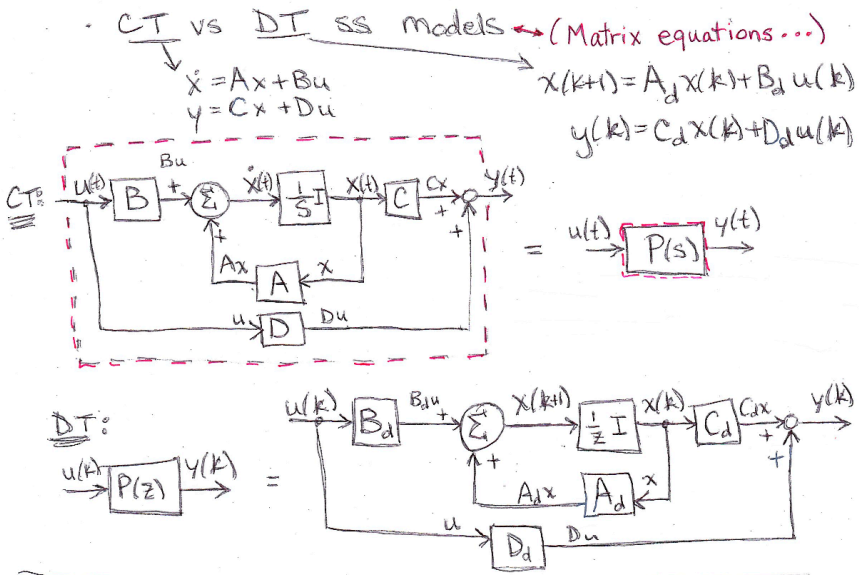
State Space: Introduction

Often, one can use MULTIPLE definitions of the "states", and of course it doesn't really matter what order you list them in "X"… Just be CONSISTENT!

e.g., same "toy example 2": (2nd-order system). Let's swap x1 and x2 within vector X:



State Space: CT vs DT Block Diagram Form



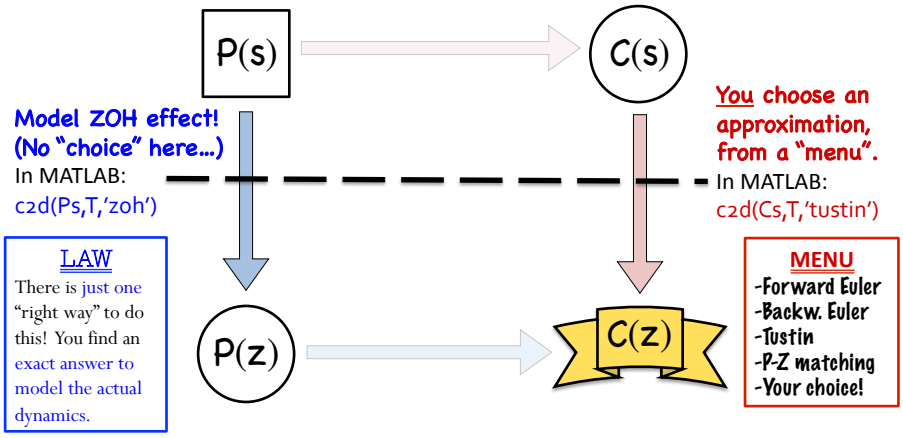
State Space: CT vs DT Block Diagram Form

STABILITY:

CT (continuous time)	DT (discrete time)

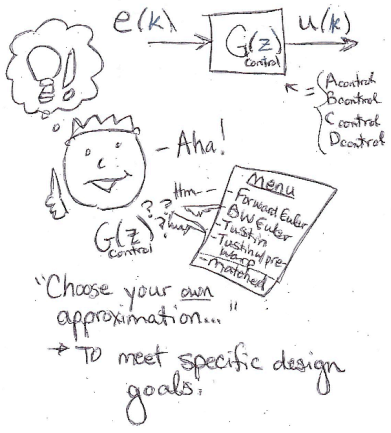
Recall we use Z xforms in 2 ways...

- There are two DIFFERENT ways we will represent dynamics via DT SS models:

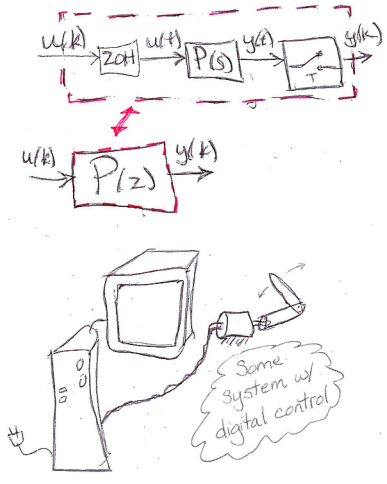


SS Representation: Again, we have 2 cases!!!

Case 1:



Case 2:



Case 1: Numeric Integration

A) Forward Euler [by far the easiest approximation, when done by hand!]

Case 1: Numeric Integration (cont...)

B) Backward Euler: A little trickier to implement for state space...

Case 1: Numeric Integration (cont...)

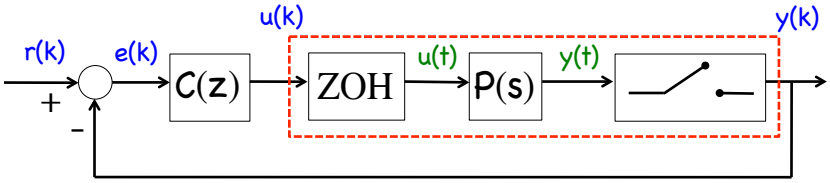
C) Tustin: Usually a good choice! (Also called "trapezoid" or "bilinear" method.)

Case 1: Numeric Integration (cont...)

SUMMARY. Stability of poles on z-plane. Table of matrix conversions.

Forward Backward Tustin/trapezoid/bilinear

Case 2: Modeling the ZOH plus sampling



Case 2: Modeling the ZOH plus sampling

Note: Analogous approximation trick applies for "e^{AT}" when "AT" has "small" entries as for "e^{sT}" when "sT" has small magnitude: