Introduction to State Space techniques

Tobin Fricke

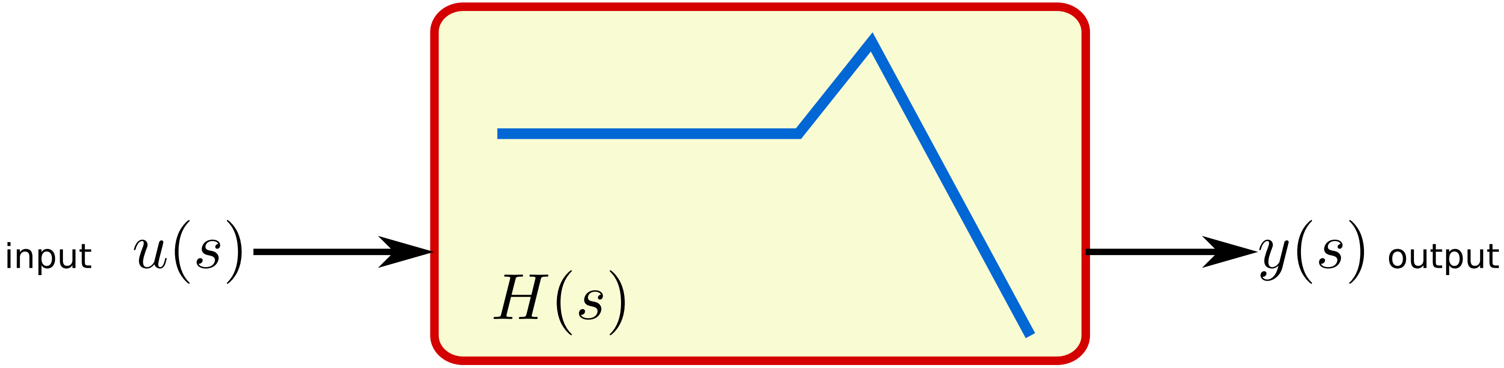
GEO ISC meeting - December 2012

tobin.fricke @ aei.mpg.de

LIGO-G1201272

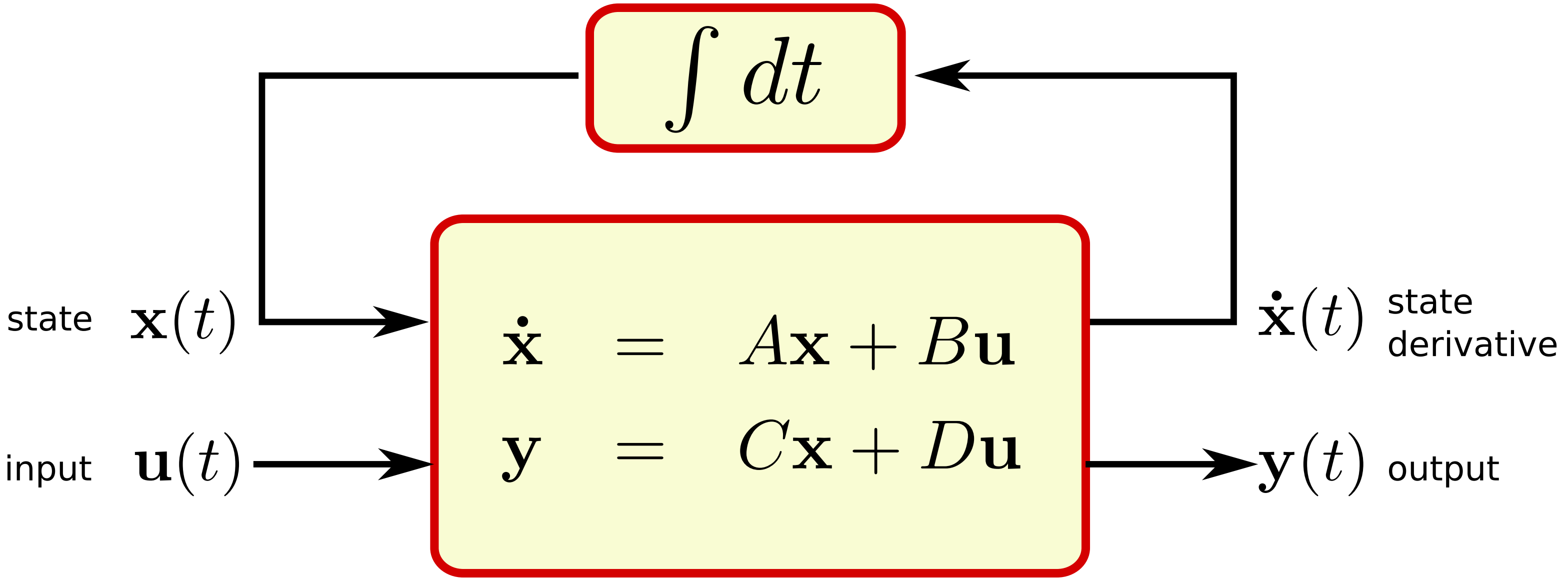
Albert Einstein Institute
Hannover

our friend the frequency domain



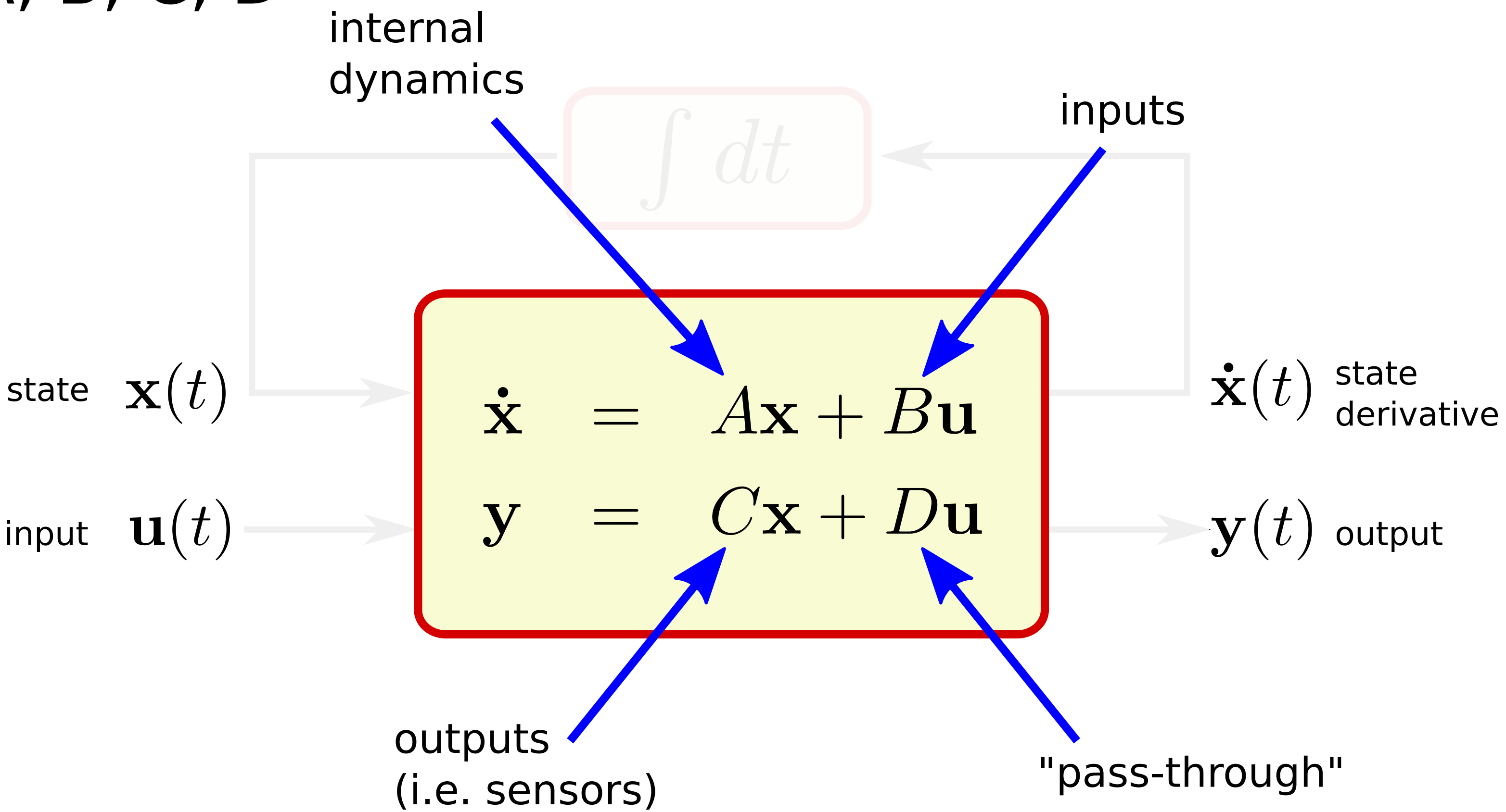
System represented as a
linear transfer function

state space



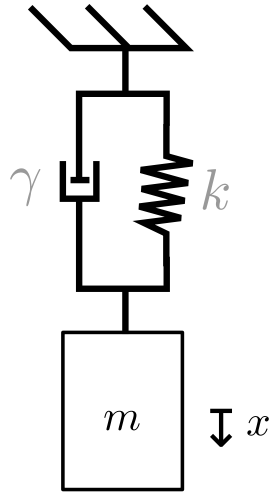
System represented as a collection of
coupled linear first-order differential equations.

A, B, C, D


internal dynamics
inputs
state $\mathbf{x}(t)$
input $\mathbf{u}(t)$
$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$
$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$
$\int dt$
$\dot{\mathbf{x}}(t)$ state derivative
$\mathbf{y}(t)$ output
outputs (i.e. sensors)
"pass-through"

an example

$$\overbrace{F_{\text{ext}} - kx - \gamma\dot{x}}^{F} = \overbrace{m\ddot{x}}^{ma}$$



system state = {position, velocity}

$$\underbrace{\begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -\gamma/m \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} x \\ \dot{x} \end{pmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{B} \underbrace{F_{ext}}_{\mathbf{u}}$$

$$\underbrace{\begin{pmatrix} y \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \underbrace{\begin{pmatrix} x \\ \dot{x} \end{pmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D} \underbrace{F_{ext}}_{\mathbf{u}}$$

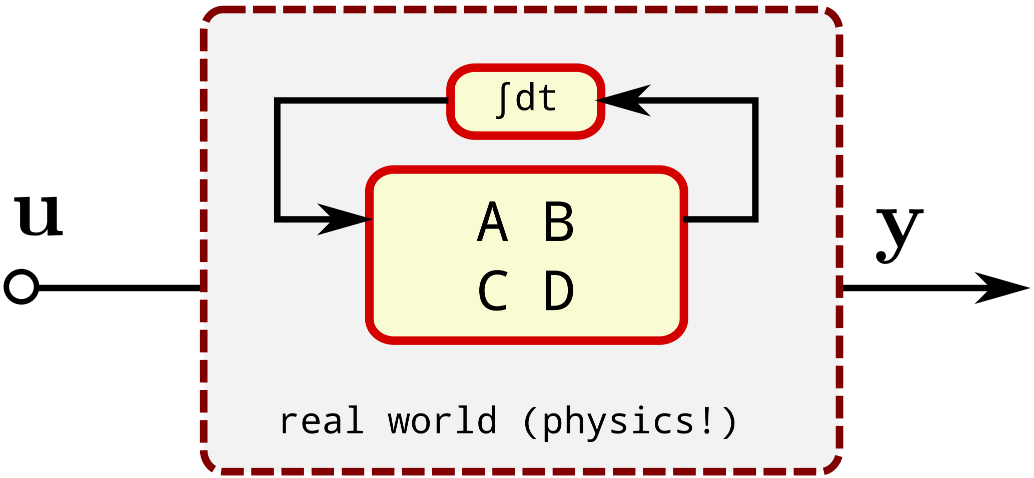
state estimation

Problem:
Not all state information is directly observable.

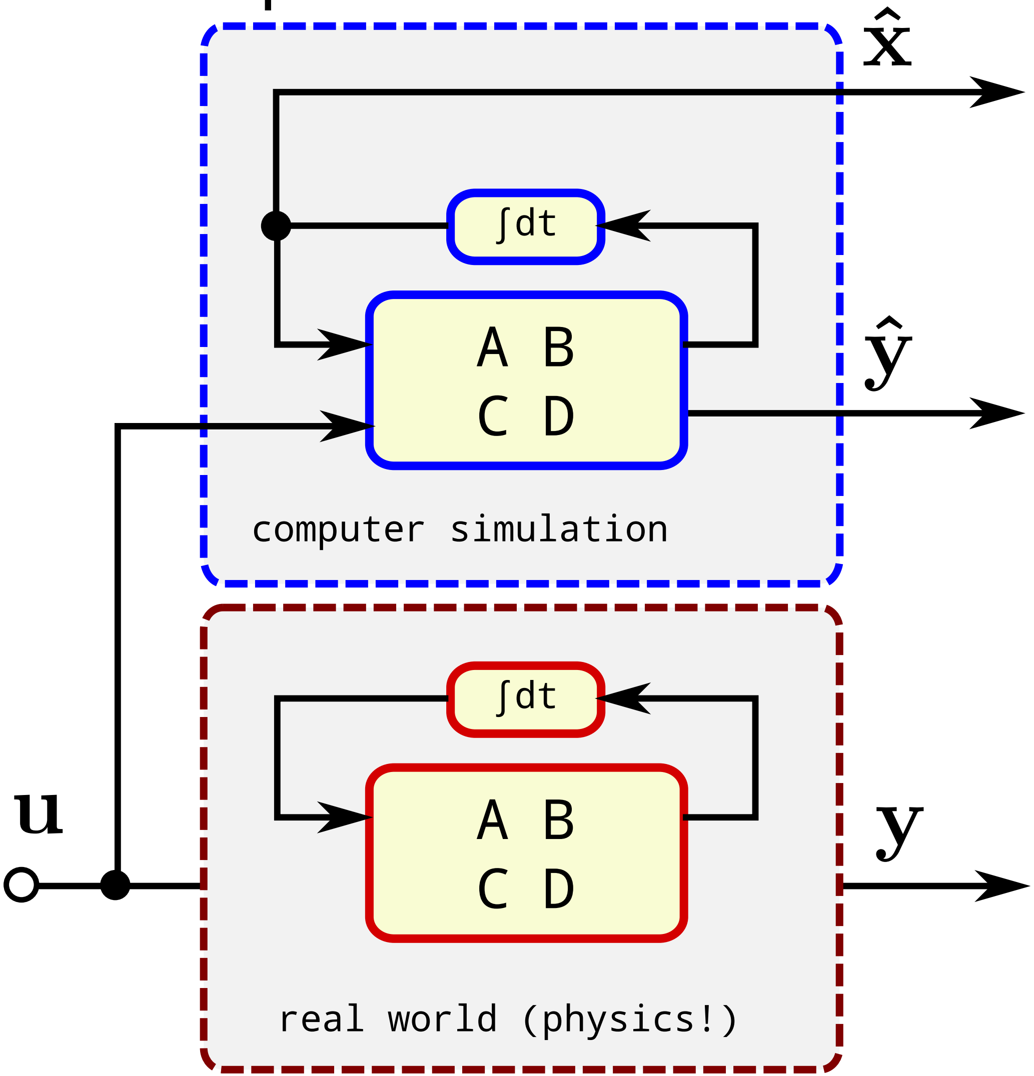
Example:
In a double suspension, you might only have sensors on one of the suspended masses.

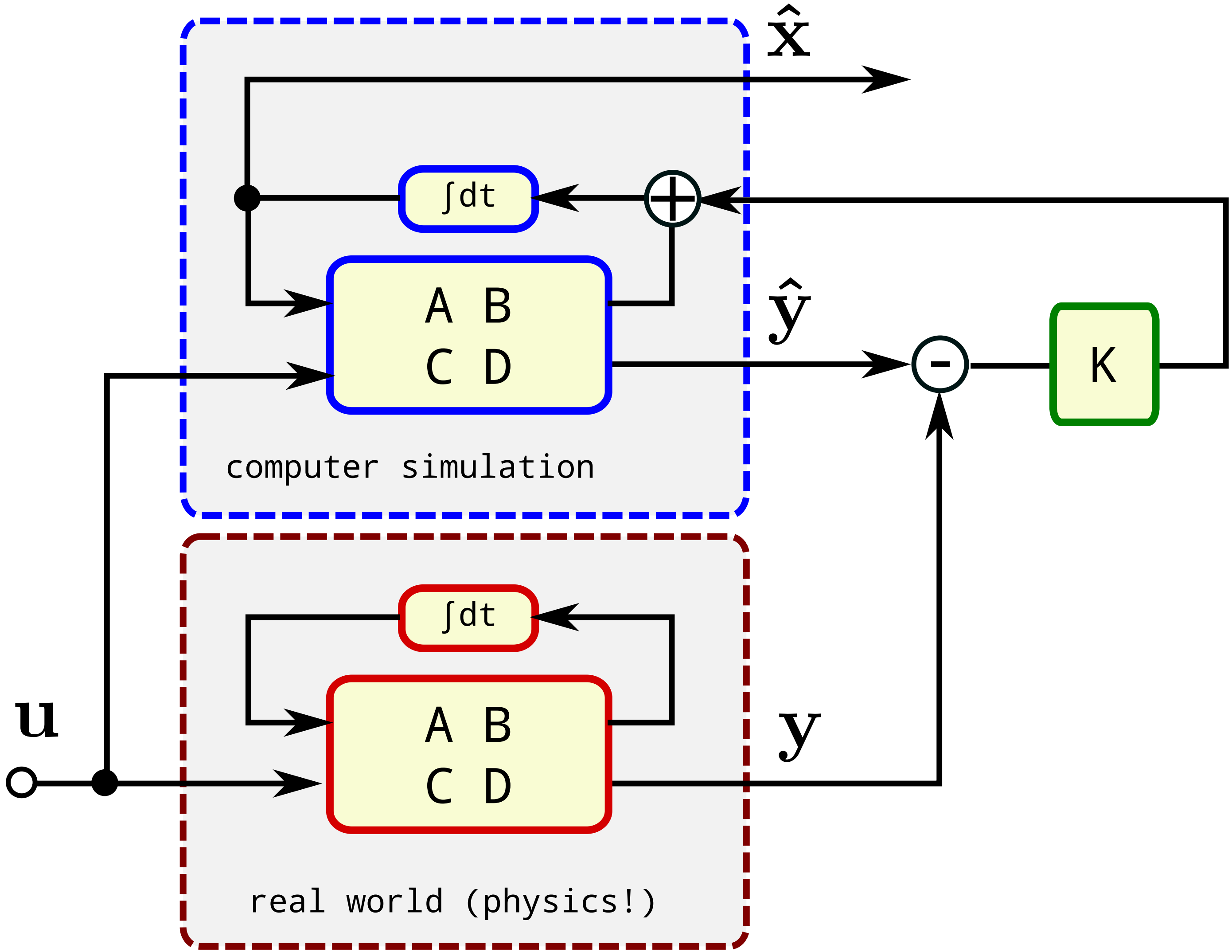
Solution:
It turns out that there is a systematic way to estimate unsensed states.

structure of state feedback...

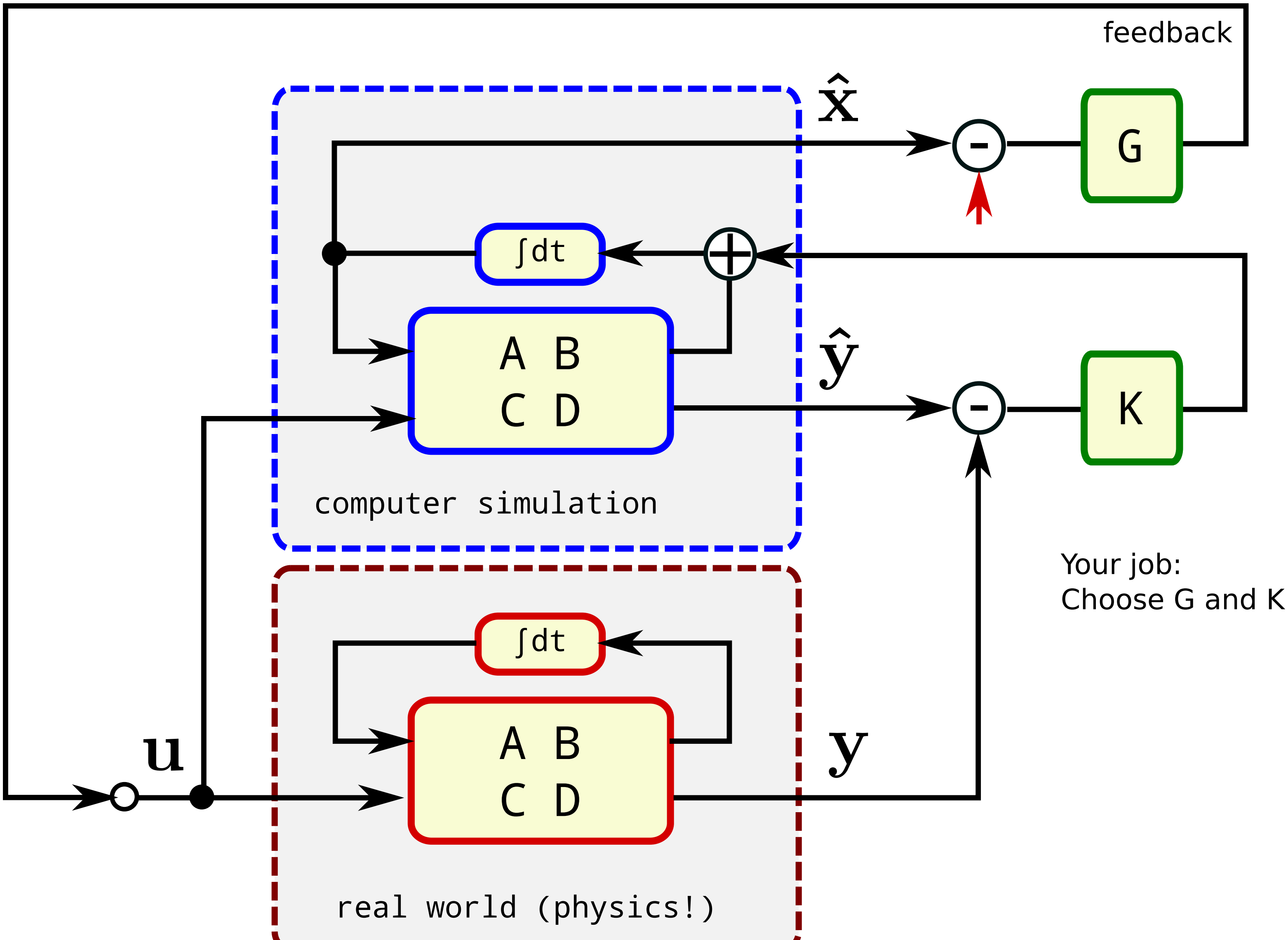


simulate in parallel





x̂
∫dt
A B
C D
ŷ
K
computer simulation
∫dt
u
A B
C D
y
real world (physics!)



feedback
x̂
G
∫dt
A B
C D
ŷ
K
computer simulation
Your job:
Choose G and K
∫dt
u
A B
C D
y
real world (physics!)

robustness

Problem:

If your model does not match your system exactly,
the resulting state estimator and controller might not work.

(See paper by Strain and Shapiro[1],
and Fu's talk at the last GEO ISC meeting!)

Solution?

This seems to be an area of active research: "**robust control**".

[1] LIGO-P1200009-v1 http://dx.doi.org/10.1063/1.4704459

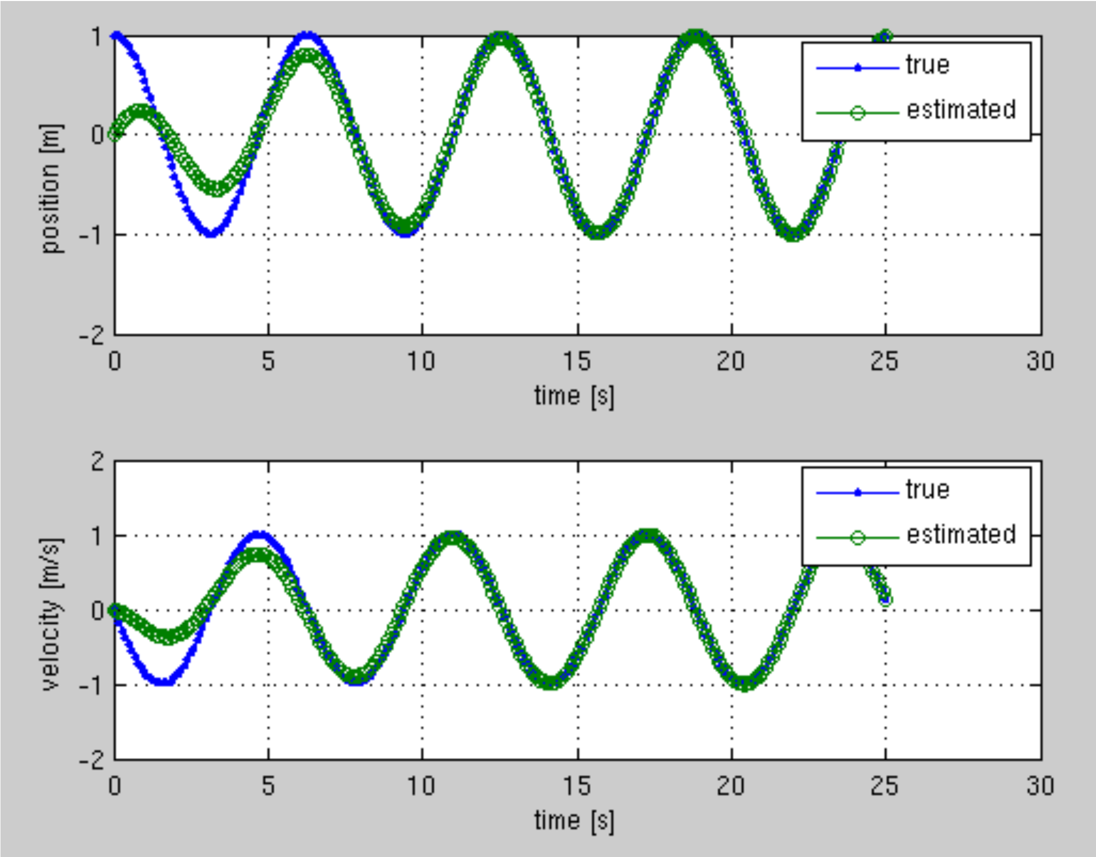
Toy example

```matlab
% State-Space Simple Harmonic Oscillator
%
% This script simulates a damped simple harmonic oscillator (i.e. mass on a
% spring) using a state-space description, and also implements a linear
% observer.
%
% Tobin Fricke
% 2012-04-13

% Parameters

k = 1;  % spring constant [m/s]
m = 1;  % mass [kg]
b = 0;  % damping constant [kg/m/s]

% Define the state-space matrices

% The dynamic system is defined by these matrices:
%
% (d/dt) x = A x + B u
%        y = C x + D u
%
% where
%
% x: internal state
% u: input to system
% y: output from system

A = [0 1; (-k/m) (-b/m)];   % state --> state derivative
B = [0 ; (1/m)];            % input --> state derivative
C = [1 0];                  % state --> output
D = 0;                      % input --> output

% Define the gains matrix for the observer

K = [0.5; -0.1];            % residual --> state hat dot

% Initial state

x = [1 ; 0];                % initial state
xhat = [0; 0];              % initial state estimate
t = 0;                      % initial time [s]
u = 0;                      % initial input
```

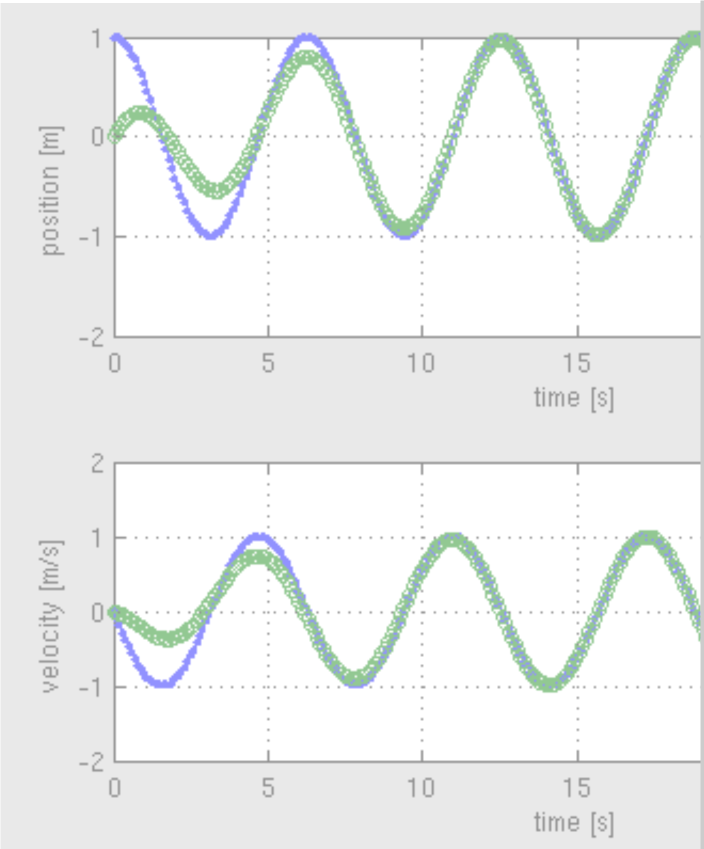
```matlab

% Simulation parameters

N = 251;                    % number of time steps
dt = 0.1;                   % time step [seconds]

% allocate space for the results

xs = nan(2, N);             % time series of true state
xhats = nan(2, N);          % time series of estimated state
ts = nan(1, N);             % time

% Do the simulation

for ii=1:length(xs)
  % record the state
  xs(:, ii) = x;
  xhats(:,ii) = xhat;
  ts(ii) = t;

  % get the current outputs
  y       = C*x + D*u;      % output of physical system
  yhat    = C*xhat + D*u;   % expected output
  r       = y - yhat;       % residual

  % advance the state
  x     = expm(A*dt)*(x + B*u*dt);
  xhat  = expm(A*dt)*(xhat + B*u*dt + K*r*dt);
  t     = t + dt;

end

% Plot the results

figure(1);
subplot(2,1,1);
plot(ts, xs(1,:), '.-', ts, xhats(1,:), 'o-');
legend('true', 'estimated');
ylabel('position [m]');
xlabel('time [s]');
grid on

subplot(2,1,2);
```

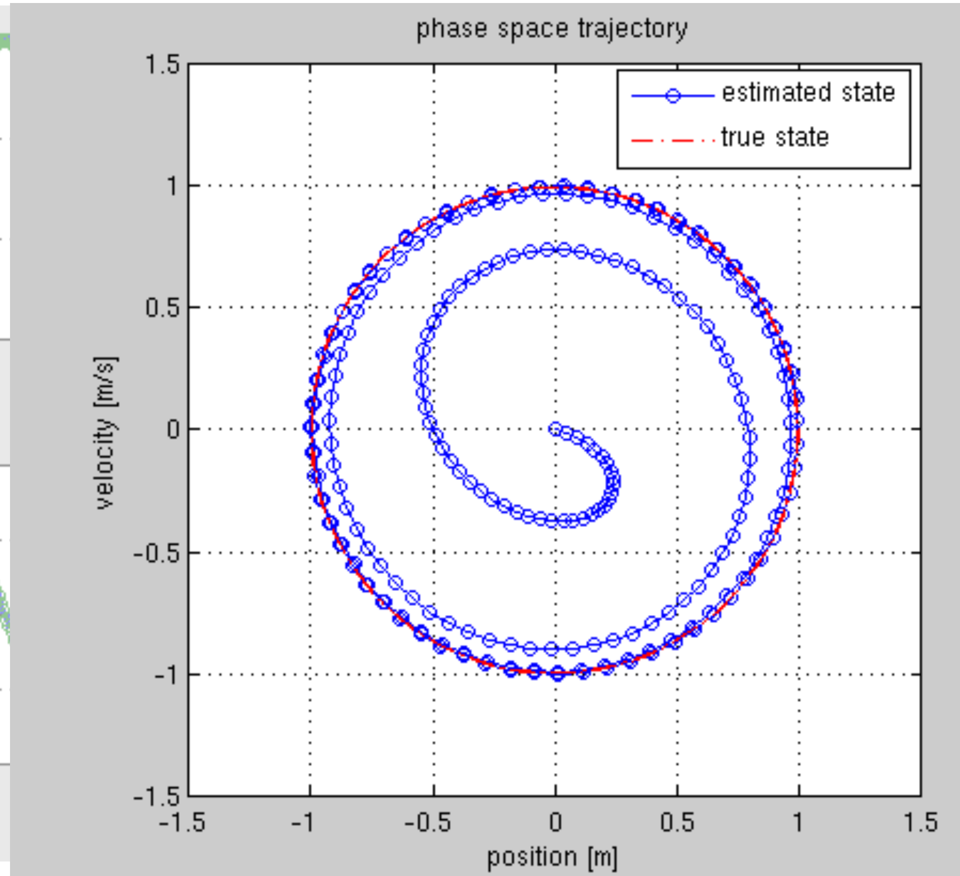
https://github.com/tobin/statespace-intro-talk

Toy example: time series



Toy example: phase plane





word cloud

step response
convolution
time domain
PID controllers
settling time
overshoot
impulse response

Bode plots
steady-state
frequency domain
transfer functions
swept-sine
Nyquist diagram
Higher-order differential equations
poles and zeroes
Linear systems
single-input,
single-output
Wiener filters
initial LIGO

Initial conditions
controllability
and observability
state estimation
internal state
nonlinear systems
multi-input,
multi-output
(MIMO)
state space
first-order differential equations
Kalman filters
the moon landing
optimum control

Questions I have

? **Is this useful to us?**
(My guesses:
for single cavities, no.
For suspensions, sometimes.
For seismic isolation - maybe?
for interferometer lock acquisition - could be very interesting!)

? **How is system identification done in state space?**
Our measurements are made in the time domain (step response) or frequency domain (transfer functions). How do we use these measurements to improve the state space models?

What could we accomplish?

Thanks for your attention!

References

Control system design: an introduction to state-space-methods
by Bernard Friedland

"How to do state space modeling"
Note by Peter Nelson, on GEO-ISC wiki