CDS 101: Lecture 3.1
Stability and Performance

Richard M. Murray

13 October 2003

Goals:

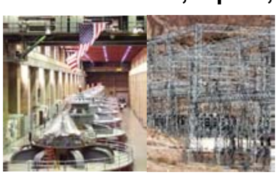
- Describe different types of stability for an equilibrium point
- Explain the difference between local/global stability, and related concepts
- Describe performance measures for (controlled) systems, including transients and steady state response

Reading:

- Åström and Murray, *Analysis and Design of Feedback Systems,* Ch 3

Review from Last Week

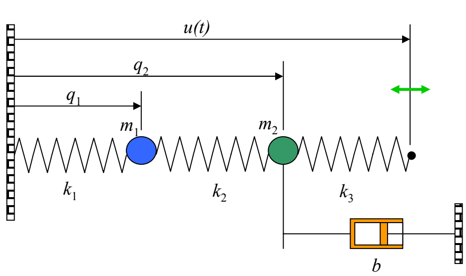
Model = state, inputs, outputs, dynamics



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

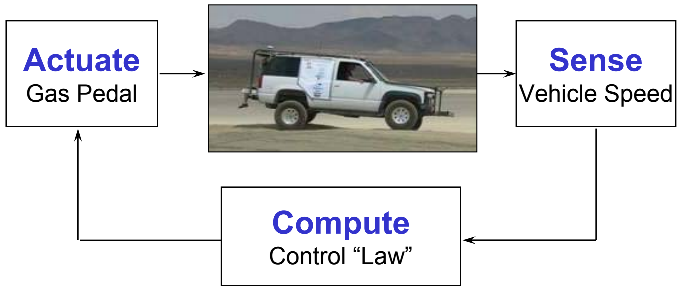
$$x_{k+1} = f(x_k, u_k)$$
$$y_{k+1} = h(x_{k+1})$$

***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 ];
```

Today: Stability and Performance



Goal #1: Stability

- Check if *closed loop* response is stable

Goal #2: Performance

- Look at ability to track changes in reference and reject disturbances

Goal #3: Robustness (later)

$\dot{x} = f(x,u) \qquad u = k(x)$

$u = k(x)$: control law; u: system input

$\dot{x} = f(x, k(x,r), d)$

$y = h(x)$

d: disturbance; r: reference; y: measured output

Systems of Ordinary Differential Equations (ODEs)

Systems of coupled nonlinear eqns (closed loop ⇒ no inputs for now)

Component form

$$\dot{x}_1 = f_1(x_1, x_2, \ldots, x_n)$$
$$\dot{x}_2 = f_1(x_1, x_2, \ldots, x_n)$$
$$\ldots$$
$$\dot{x}_n = f_1(x_1, x_2, \ldots, x_n)$$

Vector form

$$\dot{x} = f(x)$$
$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$$

MATLAB form

```
function dxdt = f(t,x)
dxdt= [
  f1(x);
  f2(x);
  …
  fn(x);
];
```

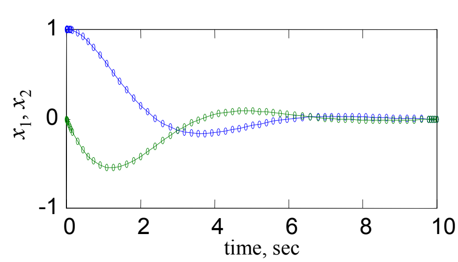
Example: damped oscillator

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_1 - x_2$$

$$f(x) = \begin{bmatrix} x_1 \\ -x_1 - x_2 \end{bmatrix} \in \mathbb{R}^2$$

```
ode45('dosc', [0 10], [1 0]);
```

function — time interval — initial condition



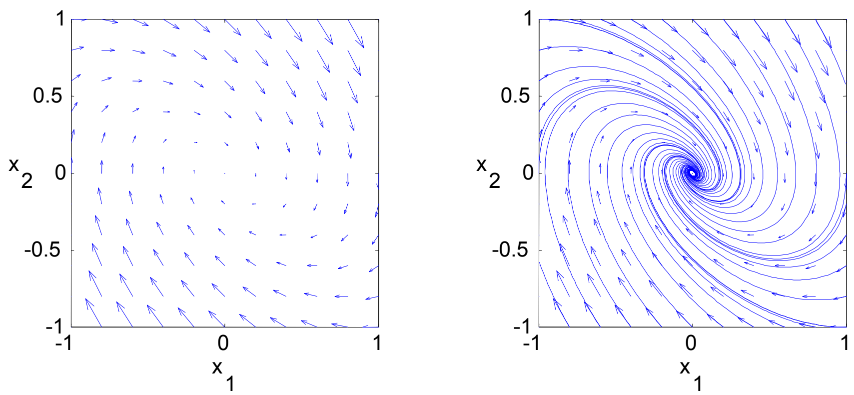
Phase Portraits (2D systems only)

Phase plane plots show 2D dynamics as *vector fields* & *stream functions*

- Plot $f(x)$ as a vector on the plane; stream lines follow the flow of the arrows

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 - x_2 \end{bmatrix}$$

```
phaseplot('dosc', ...
  [-1 1 10], [-1 1 10], ...
  boxgrid([-1 1 10], [-1 1 10]));
```

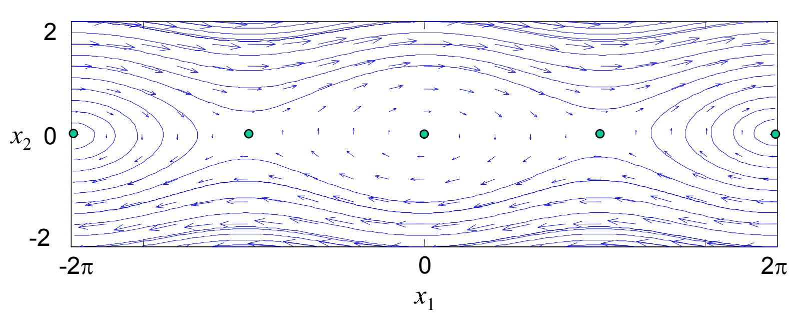


Equilibrium Points

Equilibrium points represent stationary conditions for the dynamics

The *equilibria* of the system $\dot{x} = f(x)$ are the points x_e such that $f(x_e) = 0$.

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \sin(x_1) \end{bmatrix} \quad \Rightarrow \quad x_e = \begin{bmatrix} 0 \\ \pm n\pi \end{bmatrix}$$



Stability of Equilibrium Points

An equilibrium point is:

Asymptotically stable if all nearby initial conditions converge to the equilibrium point

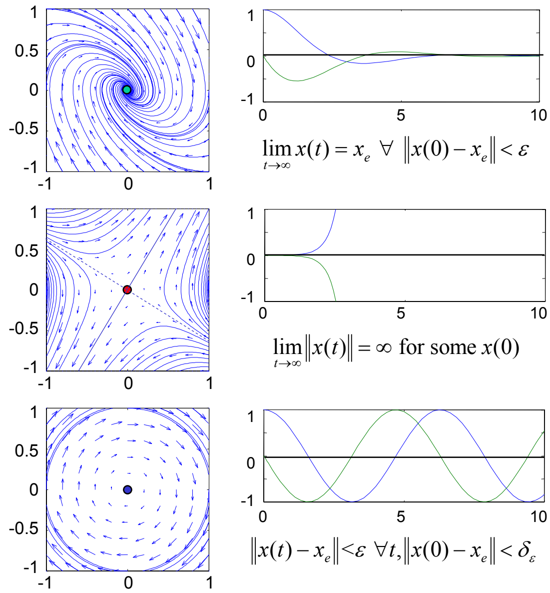
- Equilibrium point is an *attractor* or *sink*

Unstable if some initial conditions diverge from the equilibrium point

- Equilibrium point is a *source* (or *saddle*)

Stable if initial conditions that start near the equilibrium point, stay near

- Equilibrium point is a *center*



$$\lim_{t\to\infty} x(t) = x_e \;\; \forall \;\; \|x(0) - x_e\| < \varepsilon$$

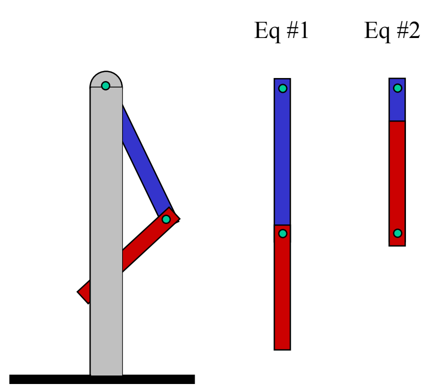
$$\lim_{t\to\infty} \|x(t)\| = \infty \text{ for some } x(0)$$

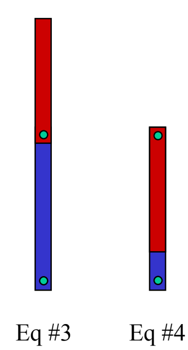
$$\|x(t) - x_e\| < \varepsilon \;\; \forall t, \|x(0) - x_e\| < \delta_\varepsilon$$

Example #1: Double Inverted Pendulum

Two series coupled pendula

- States: pendulum angles (2), velocities (2)
- Dynamics: *F* = *ma* (balance of forces)
- Dynamics are very nonlinear





Stability of equilibria

- Eq #1 is stable
- Eq #3 is unstable
- Eq #2 and #4 are unstable, but with some stable “modes”

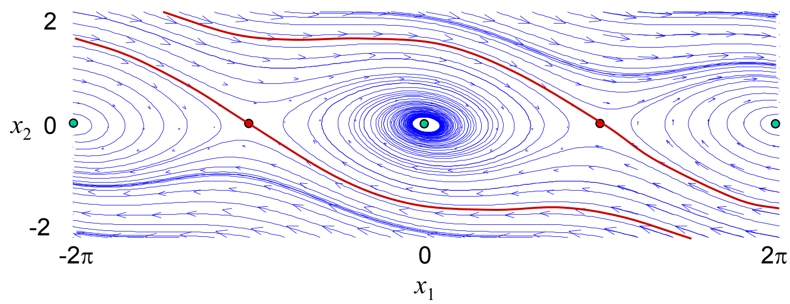
Local versus Global Behavior

Stability is a *local* concept

- Equilibrium points define the local behavior of the dynamical system
- Single dynamical system can have stable *and* unstable equilibrium points

Region of attraction

- Set of initial conditions that converge to a given equilibrium point



Example #2: Predator Prey (ODE version)

Continuous time (ODE) version of predator prey dynamics:

$$\dot{x}_1 = b_r x_1 - a x_1 x_2$$
$$\dot{x}_2 = a x_1 x_2 - d_f x_2$$

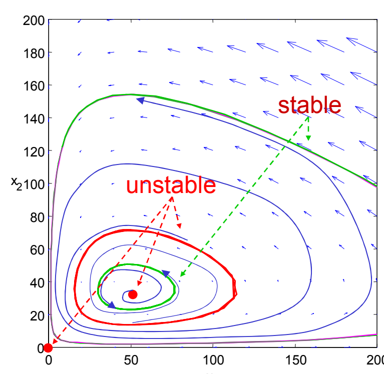
- Continuous time (ODE) model
- MATLAB: predprey.m (from web page)

Equilibrium points (2)

- (0,0): unstable ⇒ species don't die out
- ~(50,35): unstable ⇒ no steady state population

Invariant curves (3)

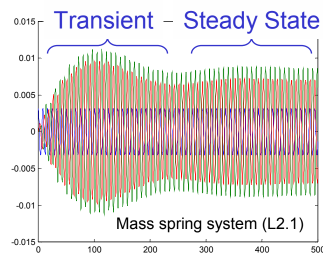
- Start on curve, stay on curve
- "Limit cycle" ⇒ population of each species oscillates over time
- This is a *global* feature of the dynamics (not local to an equilibrium point)



Input/Output Performance

Return to system with inputs
- How does system response to changes in input values?

Transient response:
- What happens right after a new input is applied



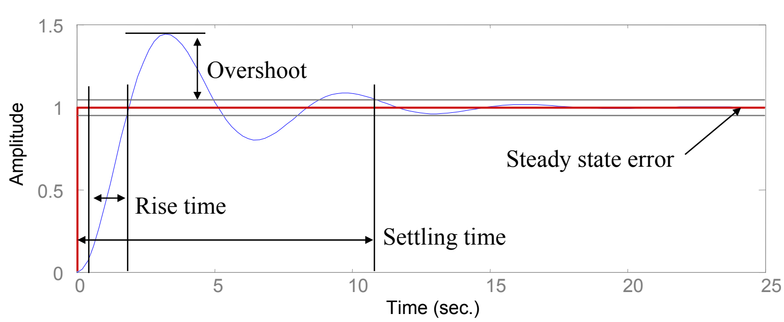
Steady state response:
- What happens a long time after the input is applied

Stability vs input/output performance
- Systems that are close to instability typically exhibit poor input/output performance
- Nearly unstable systems (slow convergence) often exhibit “ringing” (highly oscillatory response to [non-periodic] inputs)

CDS 101, 13 Oct 03 R. M. Murray, Caltech CDS 11

Step Response

Output characteristics in response to a “step” input
- Rise time: time required to move from 10% to 90% of final value
- Overshoot: ratio between amplitude of first peak and steady state value
- Settling time: time required to remain w/in *p*% (usually 2%) of final value
- Steady state error: residual error at t = ∞

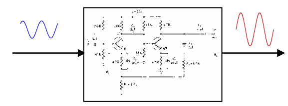


CDS 101, 13 Oct 03 R. M. Murray, Caltech CDS 12

Frequency Response

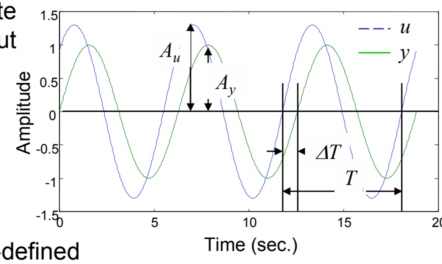
Measure the *steady state* response of the system to sinusoidal input

- Example: audio amplifier – would like consistent ("flat") amplification between 20 Hz & 20,000 Hz
- Individual sinusoids are good *test signals* for measuring performance in many systems (eg, seasonal cycles in temperature)



Approach: plot input and output, measure *relative* amplitude and phase

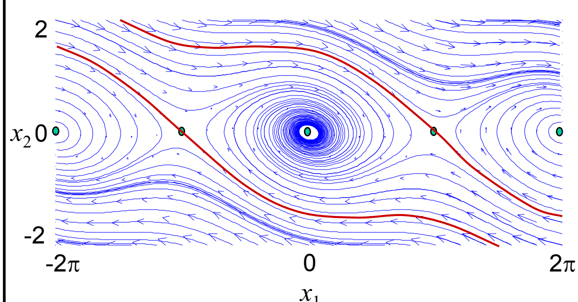
- Use MATLAB or SIMULINK to generate response of system to sinusoidal output
- Gain = A_y/A_u
- Phase = $2\pi \cdot \Delta T/T$



May not work for *nonlinear* systems

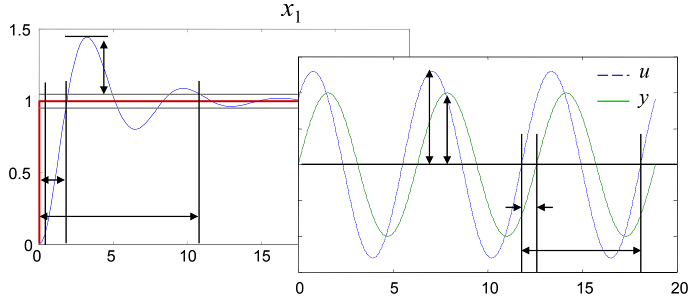
- System nonlinearities can cause *harmonics* to appear in the output
- Amplitude and phase may not be well-defined
- For *linear* systems, frequency response is always well defined (week 6)

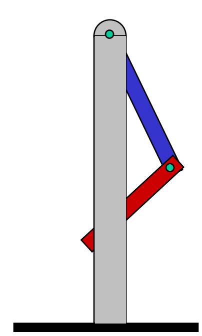
Summary: Stability and Performance



Key topics for this lecture

- Stability of equilibrium points
- Local versus global behavior
- Performance specification via step and frequency response





```
% L3_1_stability.m - MATLAB source code for Lecture 3.1
% RMM, 11 Oct 03
%
% Required files: oscillator.m, pendulum.m, saddle.m, predprey.m

%
% Systems of ODEs: damped oscillator example (simulation + phase portrait)
%
% This example uses the ODEs in dampedosc.m, available separately.

% Simulation of the damped oscillator
figure(1); ode45('oscillator', [0 10], [1 0]);
xlabel('x_1'); ylabel('x_2','Rotation',0);

% Generate a vector plot for the damped oscillator
figure(2); phaseplot('oscillator', [-1 1 10], [-1 1 10]);

% Generate a phase plot for the damped oscillator
figure(3); phaseplot('oscillator', ...
  [-1 1 10], [-1 1 10], boxgrid([-1 1 10], [-1 1 10]));

%
% Equilibrium points
%
% This example uses the undamped pendulum to illustrate multiple equilibrium
% points.
%

m = 1; l = 1; b = 0; g = 1;              % parameters for undamped pendulum
figure(1); phaseplot('pendulum', ...
   [-2*pi 2*pi 20], [-2 2 10], ...
   boxgrid([-2*pi 2*pi 20], [-2 2 10]), 10, ...
   m, l, b, g);

%
% Stability definitions
%
% This set of plots illustrates the various types of equilibrium points.
%

%
% Asy stable
%
m = 1; b = 1; k = 1;                     % default values
figure (1); phaseplot('oscillator', [-1 1 10], [-1 1 10], ...
  boxgrid([-1 1 10], [-1 1 10]), 10, m, b, k);
print -dmeta asystable.wmf

[x,t] = ode45('oscillator', [0 10], [1 0], [], m, b, k);
figure(2); plot(x, t);

%
% Unstable
%
% Note: parameter values need tweaking to generate plot from lecture
%
figure(1); phaseplot('saddle', [-1 1 10], [-1 1 10], ...
  boxgrid([-.1 .1 10], [-.1 .1 10]), 1);

[x,t] = ode45('saddle', [0 10], [1 0]);
figure(2); plot(x, t); axis([0 10 -100 100]);

%
% Stable isL
%
m = 1; b = 0; k = 1;                     % zero damping
figure (1); phaseplot('oscillator', [-1 1 10], [-1 1 10], ...
  boxgrid([-1 1 10], [-1 1 10]), 10, m, b, k);

[x,t] = ode45('oscillator', [0 10], [1 0], [], m, b, k);
figure(2); plot(x, t);

%
% Local versus global behavior
%
% This example uses a damped pendulum to illustrate regions of attraction
% The coloring in the powerpoint slide is done by hand (ugh)

m = 1; l = 1; b = 0.5; g = 1;            % parameters for damped pendulum
figure(1); phaseplot('pendulum', ...
  [-2*pi 2*pi 20], [-2 2 10], ...
  boxgrid([-2*pi 2*pi 20], [-2 2 20]), 10, ...
   m, l, b, g);

%
% Predator prey example
%
% The code implementing this is in the file predprey.m.

global pp_K pp_xeq;                      % declare control variables
pp_K = [0, 0]; pp_xeq = [50, 35];        % turn control off

% Generate a simulation of the system
figure(1);
ode45('predprey', [0 100], [10 10]);

figure(2); phaseplot('predprey', ...
  [0 200 10], [0 200 10], ...
  [50, 2; 50, 35; 50, 50.8; 50, 71.706], ...       % numerically determined
  50);
```

```
% oscillator.m - ODEs for a 2D oscillator
% RMM, 11 Oct 03
%
% This function gives the dynamics for a damped oscillator.  The states
% of the system are
%
%   x(1)  position
%   x(2)  velocity
%
% The default parameters for the system are given by
%
%   m = 1 mass, kg
%   k = 1 spring constant, N/m
%   b = 1 damping constant, N-sec/m
%
% This corresponds to a fairly heavily damped oscillator.

function dx=oscillator(t, x, flags, m, b, k)
if (nargin < 4) m = 1; end;
if (nargin < 5) b = 1; end;
if (nargin < 6) k = 1; end;

dx=[x(2); -k/m*x(1) - b/m*x(2)];
```