Time Response of Dynamic Systems

Robert Stengel
Robotics and Intelligent Systems MAE 345, Princeton University, 2017

Multi-dimensional trajectories
Numerical integration
Linear and nonlinear systems
Linearization of nonlinear models
LTI System Response
Phase-plane plots

Multi-Dimensional Trajectories

Position, velocity, and acceleration are vectors

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}; \quad \mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}; \quad \mathbf{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}; \quad \mathbf{j} = \begin{bmatrix} j_x \\ j_y \end{bmatrix}; \quad \mathbf{s} = \begin{bmatrix} s_x \\ s_y \end{bmatrix}$$

$$\begin{bmatrix} x(0) \\ x(t) \\ v_x(0) \\ v_x(t) \\ a_x(0) \\ a_x(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t & t^2/2 & t^3/6 & t^4/24 & t^5/120 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \ldots \end{bmatrix} \begin{bmatrix} x(0) \\ v_x(0) \\ a_x(0) \\ \ldots \end{bmatrix}$$

$$\begin{bmatrix} y(0) \\ y(t) \\ v_y(0) \\ v_y(t) \\ a_y(0) \\ a_y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t & t^2/2 & t^3/6 & t^4/24 & t^5/120 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & t & t^2/2 & t^3/6 & t^4/24 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & t^2/2 & t^3/6 \end{bmatrix} \begin{bmatrix} y(0) \\ v_y(0) \\ a_y(0) \\ j_y(0) \\ s_y(0) \\ c_y(0) \end{bmatrix}$$

Two-Dimensional Trajectory

Solve for Cartesian components separately

***x* Component**

$$\begin{bmatrix} j_x(0) \\ s_x(0) \\ c_x(0) \end{bmatrix} = \begin{bmatrix} -60/t^3 & 60/t^3 & -36/t^2 & -24/t^2 & -9/t & 3/t \\ 360/t^4 & -360/t^4 & 192/t^3 & 168/t^3 & 36/t^2 & -24/t^2 \\ -720/t^5 & 720/t^5 & -360/t^4 & -360/t^4 & -60/t^3 & 60/t^3 \end{bmatrix} \begin{bmatrix} x(0) \\ x(t) \\ v_x(0) \\ v_x(t) \\ a_x(0) \\ a_x(t) \end{bmatrix}$$

***y* Component**

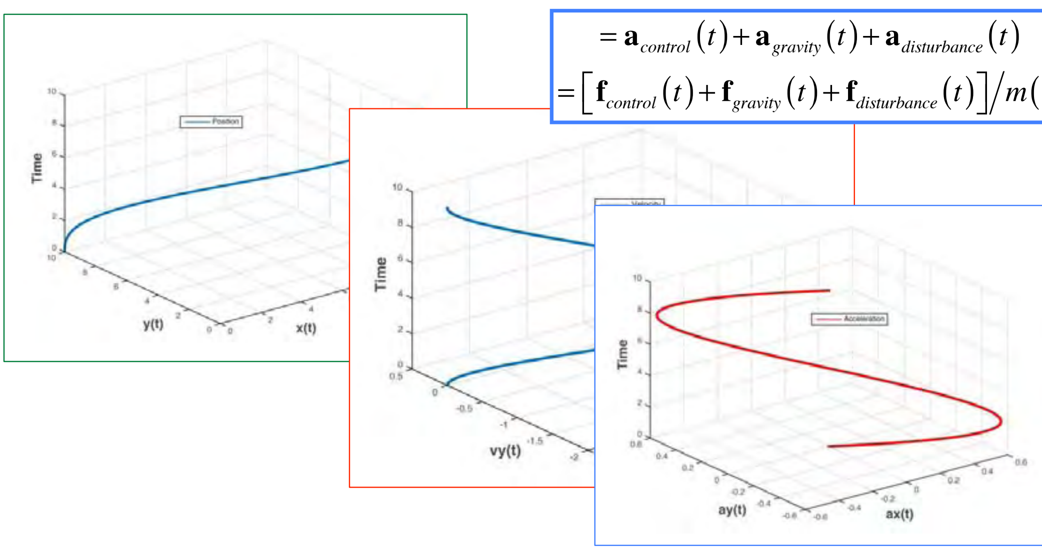
$$\begin{bmatrix} j_y(0) \\ s_y(0) \\ c_y(0) \end{bmatrix} = \begin{bmatrix} -60/t^3 & 60/t^3 & -36/t^2 & -24/t^2 & -9/t & 3/t \\ 360/t^4 & -360/t^4 & 192/t^3 & 168/t^3 & 36/t^2 & -24/t^2 \\ -720/t^5 & 720/t^5 & -360/t^4 & -360/t^4 & -60/t^3 & 60/t^3 \end{bmatrix} \begin{bmatrix} y(0) \\ y(t) \\ v_y(0) \\ v_y(t) \\ a_y(0) \\ a_y(t) \end{bmatrix}$$

Two-Dimensional Example

Required acceleration <u>vector</u> is specified by

$$\mathbf{a}(t) = \mathbf{a}(0) + \mathbf{j}(0)t + \mathbf{s}(0)t^2/2 + \mathbf{c}t^3/6$$

$$= \mathbf{a}_{control}(t) + \mathbf{a}_{gravity}(t) + \mathbf{a}_{disturbance}(t)$$
$$= \left[\mathbf{f}_{control}(t) + \mathbf{f}_{gravity}(t) + \mathbf{f}_{disturbance}(t)\right]/m(t)$$



Six-Degree-of-Freedom (Rigid Body) Equations of Motion

$$\dot{\mathbf{r}}_I = \mathbf{H}_B^I \mathbf{v}_B$$

$$\dot{\mathbf{v}}_B = \frac{1}{m}\mathbf{f}_B - \tilde{\boldsymbol{\omega}}_B \mathbf{v}_B$$

Translational position and velocity

$$\mathbf{r}_I = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \quad \mathbf{v}_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\dot{\boldsymbol{\Theta}} = \mathbf{L}_B^I \boldsymbol{\omega}_B$$

$$\dot{\boldsymbol{\omega}}_B = \boldsymbol{I}_B^{-1}\left(\mathbf{m}_B - \tilde{\boldsymbol{\omega}}_B \boldsymbol{I}_B \boldsymbol{\omega}_B\right)$$

Rotational position and velocity

$$\boldsymbol{\Theta} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}; \quad \boldsymbol{\omega}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

… nonlinear, and complex

Rate of change of Translational Position

$$\dot{x}_I = (\cos\theta\cos\psi)u + (-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi)v + (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)w$$
$$\dot{y}_I = (\cos\theta\sin\psi)u + (\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi)v + (-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi)w$$
$$\dot{z}_I = (-\sin\theta)u + (\sin\phi\cos\theta)v + (\cos\phi\cos\theta)w$$

Rate of change of Translational Velocity

$$\dot{u} = X/m - g\sin\theta + rv - qw$$
$$\dot{v} = Y/m + g\sin\phi\cos\theta - ru + pw$$
$$\dot{w} = Z/m + g\cos\phi\cos\theta + qu - pv$$

Rate of change of Angular Position

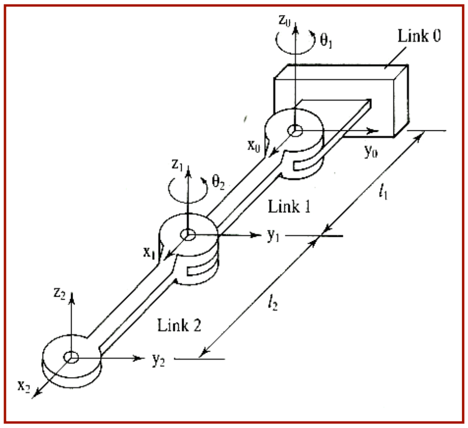
$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta$$
$$\dot{\theta} = q\cos\phi - r\sin\phi$$
$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta$$

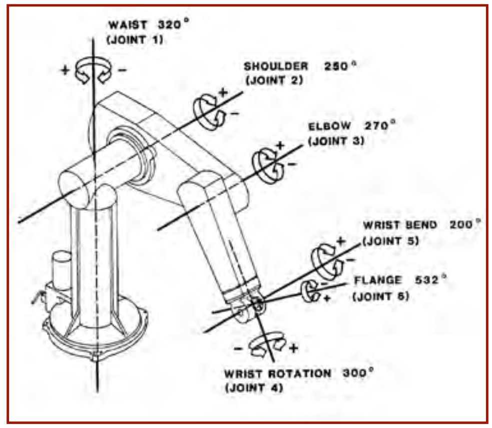
Rate of change of Angular Velocity

$$\dot{p} = \left(I_{zz}L + I_{xz}N - \left\{I_{xz}\left(I_{yy} - I_{xx} - I_{zz}\right)p + \left[I_{xz}^2 + I_{zz}\left(I_{zz} - I_{yy}\right)\right]r\right\}q\right) \Big/ \left(I_{xx}I_{zz} - I_{xz}^2\right)$$
$$\dot{q} = \left[M - \left(I_{xx} - I_{zz}\right)pr - I_{xz}\left(p^2 - r^2\right)\right] \Big/ I_{yy}$$
$$\dot{r} = \left(I_{xz}L + I_{xx}N - \left\{I_{xz}\left(I_{yy} - I_{xx} - I_{zz}\right)r + \left[I_{xz}^2 + I_{xx}\left(I_{xx} - I_{yy}\right)\right]p\right\}q\right) \Big/ \left(I_{xx}I_{zz} - I_{xz}^2\right)$$

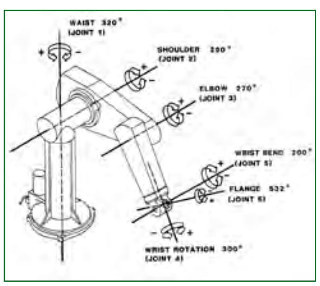
Multiple Rigid Links Lead to Multiple Constraints

- Each link is subject to the same 6-DOF rigid-body dynamic equations
- … but each link is constrained to have a single degree of freedom w.r.t. proximal link





Newton-Euler Link Dynamics



- Link dynamics are coupled
  - Proximal-link loads affected by distal-link positions and velocities
  - Distal-link accelerations affected by proximal-link motions
  - Joints produce constraints on link motions
- Net forces and torques at each joint related to velocities and accelerations of the centroids of the links
- Equations of motion derived directly for each link, with constraints

$$\dot{\mathbf{v}}_B = \frac{1}{m}\mathbf{f}_B - \tilde{\boldsymbol{\omega}}_B \mathbf{v}_B$$
$$\dot{\boldsymbol{\omega}}_B = \boldsymbol{I}_B^{-1}\left(\mathbf{m}_B - \tilde{\boldsymbol{\omega}}_B \boldsymbol{I}_B \boldsymbol{\omega}_B\right)$$

→

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t),\mathbf{u}(t),\mathbf{w}(t),\mathbf{p}(t),t\right]$$

Lagrangian Link Dynamics

- **Lagrange's equation derives from Newton's Laws**
  - **Principle of virtual work**
  - **D'Alembert's principle**
- **Dynamic behavior described by work done and energy stored in the system**
- **Equations of motion derived from *Lagrangian* function and Lagrange's equation**

$$\text{Generalized coordinate, } \mathrm{q}_n \text{, e.g., } (x, y, z, \psi, \theta, \phi)$$
$$\text{Generalized force, } \mathbf{F}_n \text{, e.g., } (f_x, f_y, f_z, m_x, m_y, m_z)$$

$$L(\mathrm{q}_n, \dot{\mathrm{q}}_n) \triangleq KE - PE$$

$$\frac{d}{dt}\left(\frac{dL(\mathrm{q}_n, \dot{\mathrm{q}}_n)}{d\dot{\mathrm{q}}_n}\right) - \frac{\partial L(\mathrm{q}_n, \dot{\mathrm{q}}_n)}{\partial \mathrm{q}_n} = \mathbf{F}_n$$

$$\rightarrow \quad \dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t\right]$$

Hamiltonian Link Dynamics

- **Hamilton's Principle: *Lagrange's equation* is a necessary condition for an extremum (i.e., maximum or minimum)**

$$\text{extremum } I = \int_{t_1}^{t_2} L(q_n, \dot{q}_n)\, dt$$

Equations of motion derived from an optimization problem

- ***Hamiltonian* function**

$$H(p, q) = \sum \dot{q}_n p_n - L(q_n, \dot{q}_n)$$

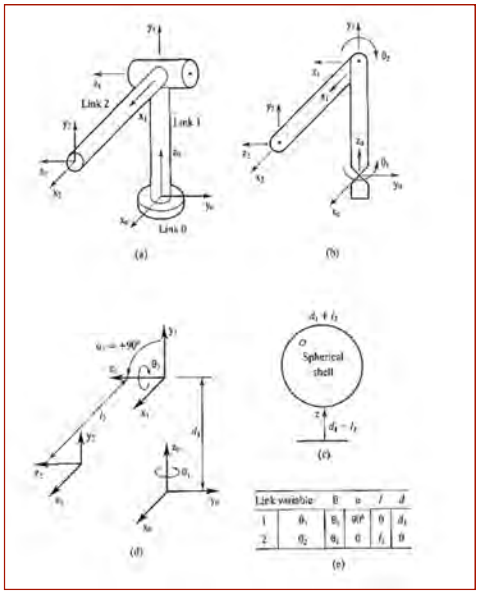
- **Hamilton's equations**

$$\text{Generalized momentum, } \mathrm{p}_n\text{, e.g., } (mv_x, mv_y, mv_z, I_{xx}p, I_{yy}q, I_{zz}r)$$

$$\dot{q}_n = \frac{\partial H(p, q)}{\partial p_n}$$
$$\dot{p}_n = -\frac{\partial H(p, q)}{\partial q_n}$$

$$\rightarrow \quad \dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t\right]$$

Example: Two-Link Robot Equations of Motion



McKerrow, 1991

State vector

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \text{Angle of } 1^{st} \text{ link}, \theta_1, \text{rad} \\ \text{Angular rate of } 1^{st} \text{ link}, \dot{\theta}_1, \text{rad/sec} \\ \text{Angle of } 2^{nd} \text{ link}, \theta_2, \text{rad} \\ \text{Angular rate of } 2^{nd} \text{ link}, \dot{\theta}_2, \text{rad/sec} \end{bmatrix}$$

Control vector

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \text{, torque at } 1^{st} \text{ joint} \\ \tau_2 \text{, torque at } 2^{nd} \text{ joint} \end{bmatrix}$$

When Possible, Simplify the Equations

Two-link robot equations of motion

- Mass, ***m***, located at end of Link 2
- Inertias of Links 1 and 2 neglected

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{\cos^2 x_3}\left(x_2 x_4 \sin 2x_3 + \frac{u_1}{m l_2^{\,2}} \right)$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -\frac{g}{l_2}\cos x_3 - \frac{x_2}{2}\sin 2x_3 + \frac{u_2}{m l_2^{\,2}}$$

from Example 7.1, Lagrangian derivation, McKerrow, 1991, pp. 388-390.

Differential Equations Integrated to Produce Time Response

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t\right], \quad \mathbf{x}(0) \text{ given}$$

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t\right] dt$$

Numerical integration is an approximation

Rectangular and Trapezoidal Integration of Differential Equations

Rectangular (Euler) Integration

$$\begin{aligned}\mathbf{x}(t_k) &= \mathbf{x}(t_{k-1}) + \Delta\mathbf{x}(t_{k-1}, t_k) \\ &\approx \mathbf{x}(t_{k-1}) + \mathbf{f}\left[\mathbf{x}(t_{k-1}), \mathbf{u}(t_{k-1}), \mathbf{w}(t_{k-1})\right] \Delta t, \quad \Delta t = t_k - t_{k-1}\end{aligned}$$

Trapezoidal (modified Euler) Integration *(ode23)*

$$\mathbf{x}(t_k) \approx \mathbf{x}(t_{k-1}) + \frac{1}{2}\left[\Delta\mathbf{x}_1 + \Delta\mathbf{x}_2\right]$$

where

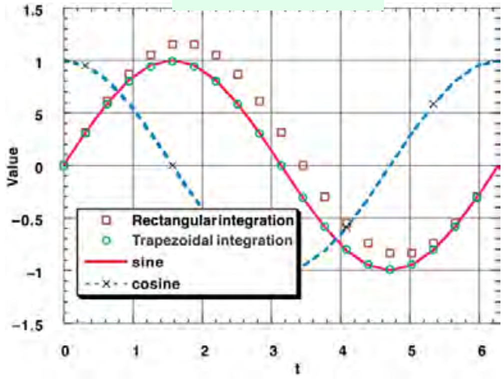
$$\Delta\mathbf{x}_1 = \mathbf{f}\left[\mathbf{x}(t_{k-1}), \mathbf{u}(t_{k-1}), \mathbf{w}(t_{k-1})\right] \Delta t$$

$$\Delta\mathbf{x}_2 = \mathbf{f}\left\{\left[\mathbf{x}(t_{k-1}) + \Delta\mathbf{x}_1\right], \mathbf{u}(t_k), \mathbf{w}(t_k)\right\} \Delta t$$

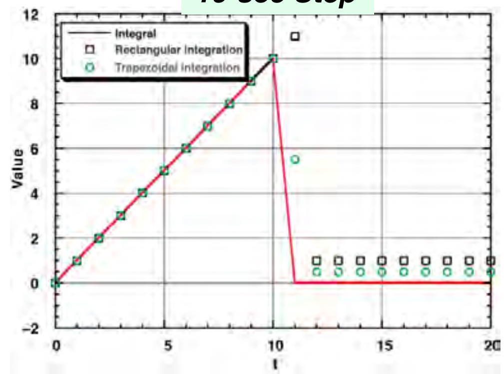
ode23 **varies step size, Δt, to reduce numerical error**

Numerical Integration Examples

Cosine Wave



10-sec Step

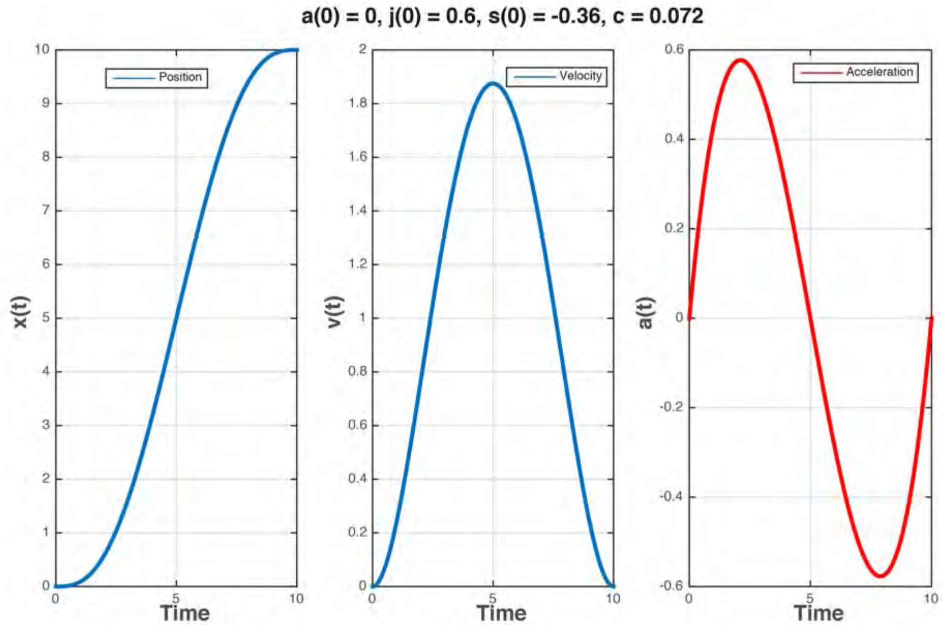


How can approximation accuracy be improved?

More Complicated Algorithms
(e.g., MATLAB)

Solver	Problem Type	Order of Accuracy	When to Use
`ode45`	Nonstiff	Medium	Most of the time. This should be the first solver you try.
`ode23`	Nonstiff	Low	For problems with crude error tolerances or for solving moderately stiff problems.
`ode113`	Nonstiff	Low to high	For problems with stringent error tolerances or for solving computationally intensive problems.
`ode15s`	Stiff	Low to medium	If `ode45` is slow because the problem is stiff.
`ode23s`	Stiff	Low	If using crude error tolerances to solve stiff systems and the mass matrix is constant.
`ode23t`	Moderately Stiff	Low	For moderately stiff problems if you need a solution without numerical damping.
`ode23tb`	Stiff	Low	If using crude error tolerances to solve stiff systems.

1-D Example



$$a_{net}(t) = (0) + 0.6t - 0.36t^2/2 + 0.072t^3/6$$

Comparison of Exact and Numerically Integrated Trajectories

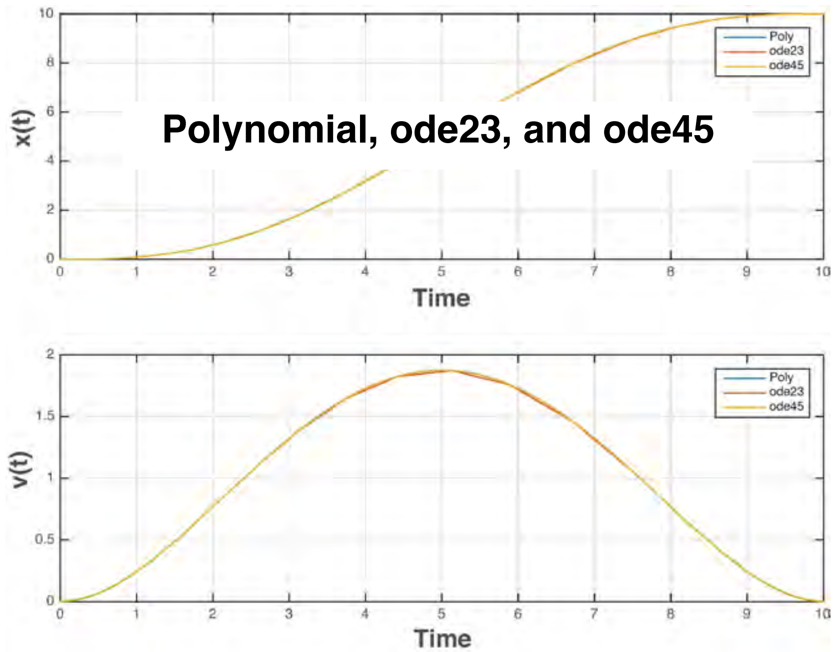
Calculate trajectory, given constants for $t_f = 10$

$$\begin{bmatrix} x(t) \\ v(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} 1 & t & t^2/2 & t^3/6 & t^4/24 & t^5/120 \\ 0 & 1 & t & t^2/2 & t^3/6 & t^4/24 \\ 0 & 0 & 1 & t & t^2/2 & t^3/6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.6 \\ -0.36 \\ 0.072 \end{bmatrix}$$

Calculate trajectory by numerical integration

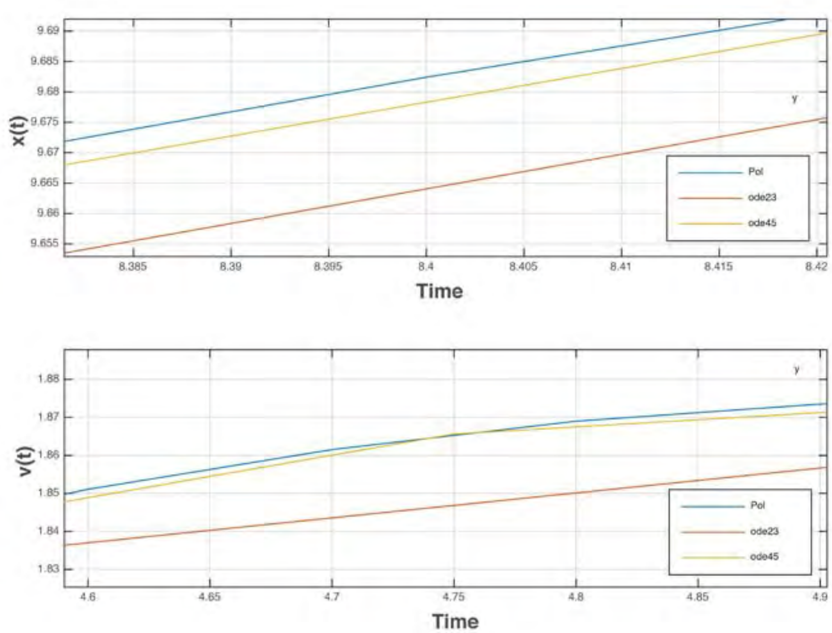
$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = \dot{v}(t) = a(t) = 0.6t - 0.36t^2/2 + 0.072t^3/6$$

Comparison of Exact and Numerically Integrated Trajectories

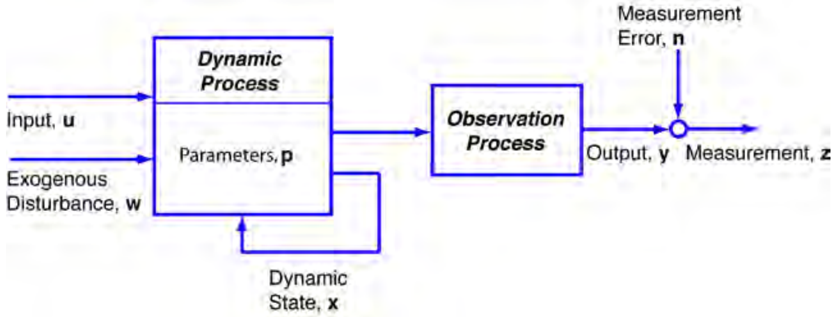


Comparison of Exact and Numerically Integrated Trajectories *(Zoom)*

Polynomial, ode23, and ode45



Generic Dynamic System



Dynamic Process: Current state may depend on prior state

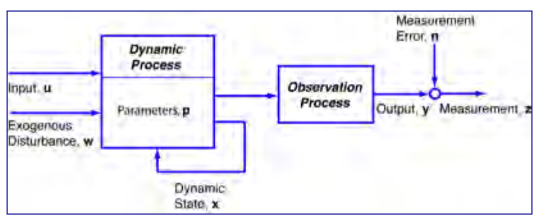
x : state — *dim* = (n x 1)
u : input — *dim* = (m x 1)
w : disturbance — *dim* = (s x 1)
p : parameter — *dim* = (ℓ x 1)

t : time (independent variable, 1 x 1)

Observation Process: Measurement may contain error or be incomplete

y : output (error-free)
dim = (r x 1)
n : measurement error
dim = (r x 1)
z : measurement
dim = (r x 1)

Equations of the System



Dynamic Equation

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t]$$

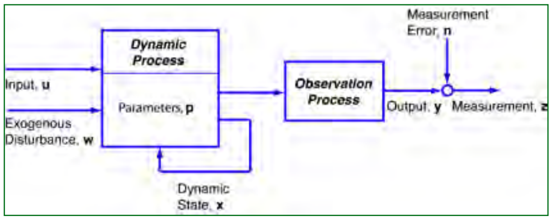
Output Equation

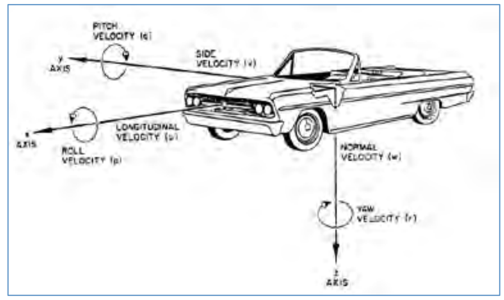
$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t)]$$

Measurement Equation

$$\mathbf{z}(t) = \mathbf{y}(t) + \mathbf{n}(t)$$

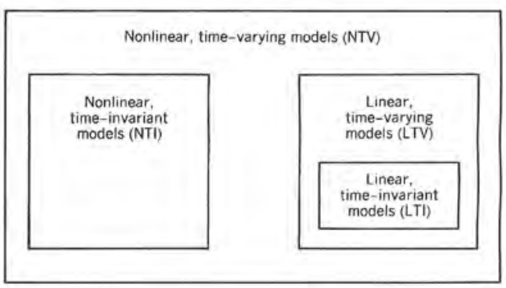
Dynamic System Example: Automotive Vehicle





- **Dynamic Process**
  - **x : dynamic state**
    - Position, velocity, angle, angular rate
  - **u : input**
    - Steering, throttle, brakes
  - **w : disturbance**
    - Road surface, wind
  - **p : parameter**
    - Weight, moments of inertia, drag coefficient, spring constants
  - ***t* : time (independent variable)**

- **Observation Process**
  - **y : error-free output**
    - Speed, front-wheel angle, engine rpm, acceleration, yaw rate, throttle, brakes, GPS location
  - **n : measurement error**
    - Perturbations to y
  - **z : measurement**
    - Sum of y and n



Nonlinearity and Time Variation in Dynamic Systems

Nonlinear, time-varying dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \mathbf{p}(t), t\right] \quad \textit{NTV}$$

Nonlinear, time-invariant dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)\right] \quad \textit{NTI}$$

Linear, time-varying dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{u}(t) + \mathbf{L}(t)\mathbf{w}(t) \quad \textit{LTV}$$

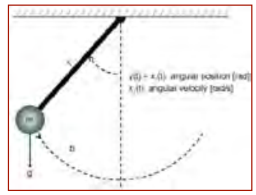
Linear, time-invariant dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t) + \mathbf{L}\mathbf{w}(t) \quad \textit{LTI}$$

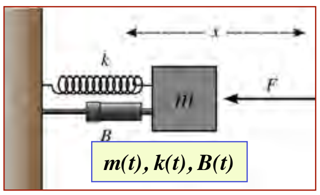
Nonlinearity and Time Variation in Dynamic Systems

Nonlinear, time-varying dynamics

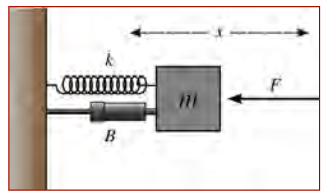
Nonlinear, time-invariant dynamics



Linear, time-varying dynamics



Linear, time-invariant dynamics



Solutions of Ordinary Differential Dynamic Equations

Time-Domain Model (ODE)	Solution by Numerical Integration	Principle of Superposition	Frequency-Domain Model
. Nonlinear, time-varying (NTV)	*Yes*	*No*	*No*
. Nonlinear, time-invariant (NTI)	*Yes*	*No*	*Yes (amplitude-dependent, harmonics)*
. Linear, time-varying (LTV)	*Yes*	*Yes*	*Approximate*
. Linear, time-invariant (LTI)	*Yes*	*Yes*	*Yes*

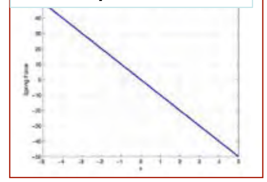
Comparison of Damped Linear and Nonlinear Systems

Linear Spring

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -10x_1(t) - x_2(t)$$

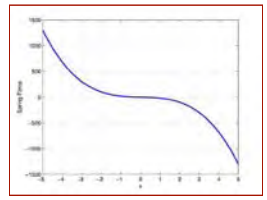
Spring **Damper**

Spring Force vs. Displacement



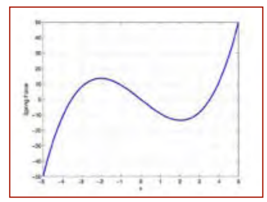
Linear plus Stiffening Cubic Spring

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -10x_1(t) - 10x_1^3(t) - x_2(t)$$



Linear plus Weakening Cubic Spring

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -10x_1(t) + 0.8x_1^3(t) - x_2(t)$$



NTV or NTI?

MATLAB Simulation of Linear and Nonlinear Dynamic Systems

MATLAB Main Program

```
% Nonlinear and Linear Examples
    clear
    tspan = [0 10];
    xo = [0, 10];
    [t1,x1 = ode23('NonLin',tspan,xo);
    xo = [0, 1];
    [t2,x2] = ode23('NonLin',tspan,xo);
    xo = [0, 10];
    [t3,x3] = ode23('Lin',tspan,xo);
    xo = [0, 1];
    [t4,x4] = ode23('Lin',tspan,xo);

    subplot(2,1,1)
    plot(t1,x1(:,1),'k',t2,x2(:,1),'b',t3,x3(:,1),'r',t4,x4(:,1),'g')
    ylabel('Position'), grid
    subplot(2,1,2)
    plot(t1,x1(:,2),'k',t2,x2(:,2),'b',t3,x3(:,2),'r',t4,x4(:,2),'g')
    xlabel('Time'), ylabel('Rate'), grid
```

Linear Spring

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -10x_1(t) - x_2(t)$$

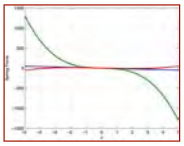
```
function xdot = Lin(t,x)
% Linear Ordinary Differential Equation
% x(1) = Position
% x(2) = Rate
xdot = [x(2)
        -10*x(1) - x(2)];
```

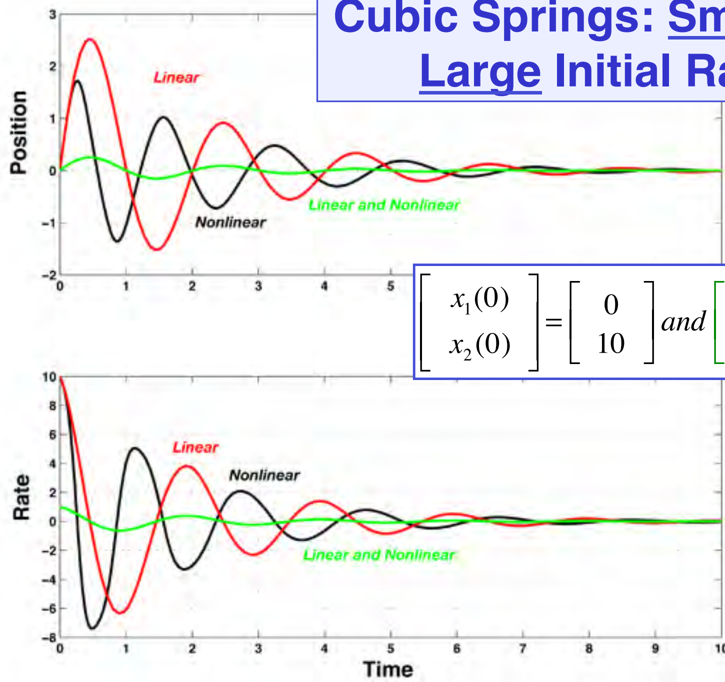
Weakening Spring

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = -10x_1(t) + 0.8x_1^3(t) - x_2(t)$$

```
function xdot = NonLin(t,x)
% Nonlinear Ordinary Differential Equation
% x(1) = Position
% x(2) = Rate
xdot = [x(2)
        -10*x(1) + 0.8*x(1)^3 - x(2)];
```

Linear and Stiffening Cubic Springs: Small and Large Initial Rates

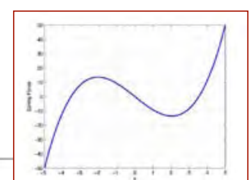


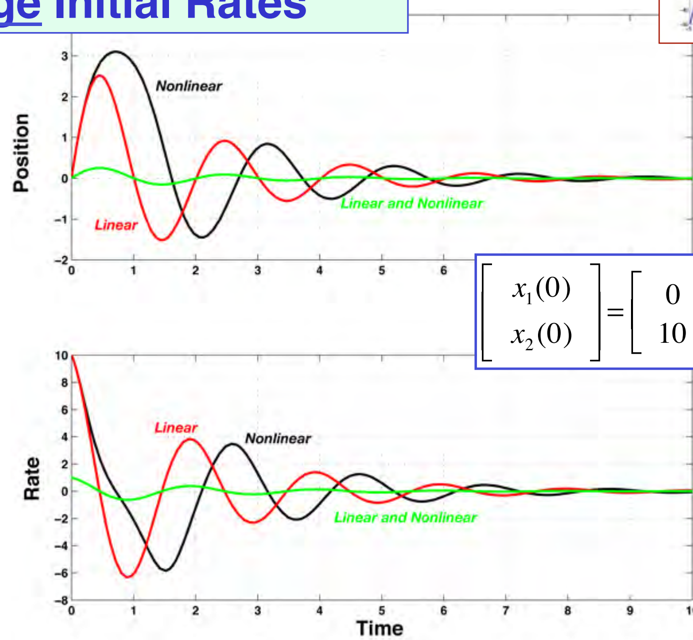


$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix} and \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Linear and nonlinear responses are indistinguishable with small initial condition

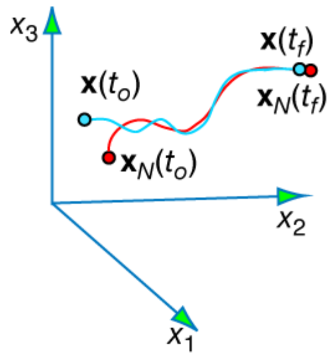
Linear and Weakening Cubic Springs: Small and Large Initial Rates





$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix} and \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Linearization of Nonlinear Equations



$$\dim(\mathbf{x}) = n \times 1$$
$$\dim(\mathbf{u}) = m \times 1$$
$$\dim(\mathbf{w}) = s \times 1$$

- Given
  - **Nominal** (or **reference**) robot trajectory, control, and disturbance histories

$$\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t) \quad \text{for } t \text{ in } [t_o, t_f]$$

  - **Actual** path and inputs

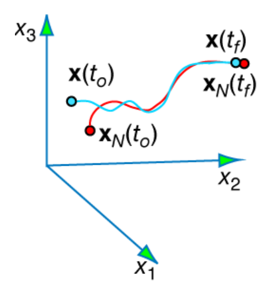
$$\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t) \quad \text{for } t \text{ in } [t_o, t_f]$$

  - perturbed by
    - Initial condition variation
    - Control variation
    - Disturbance variation

State, Control, and Disturbance Perturbations

Difference between nominal and actual paths:

$$\Delta\mathbf{x}(t_o) = \mathbf{x}(t_o) - \mathbf{x}_N(t_o)$$
$$\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_N(t)$$



Difference between nominal and actual inputs:

$$\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_N(t) \quad [\text{Control perturbation}]$$
$$\Delta\mathbf{w}(t) = \mathbf{w}(t) - \mathbf{w}_N(t) \quad [\text{Disturbance perturbation}]$$

Linear Approximation of Perturbation Effects

Nominal and actual paths both satisfy same dynamic equations

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t), t]$$
$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t]$$

Actual dynamics expressed as sum of nominal terms plus perturbations

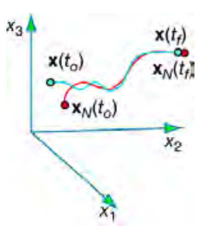
$$\begin{aligned}\dot{\mathbf{x}}(t) &= \dot{\mathbf{x}}_N(t) + \Delta\dot{\mathbf{x}}(t) \\ &= \mathbf{f}\left\{[\mathbf{x}_N(t) + \Delta\mathbf{x}(t)], [\mathbf{u}_N(t) + \Delta\mathbf{u}(t)], [\mathbf{w}_N(t) + \Delta\mathbf{w}(t), t]\right\}\end{aligned}$$

Exact

$$\approx \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t), t] + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(t)\Delta\mathbf{x}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(t)\Delta\mathbf{u}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{w}}(t)\Delta\mathbf{w}(t)$$

Approx.

Partial-derivative (*Jacobian*) matrices are evaluated along the nominal path



Linearized Equation Approximates Perturbation Dynamics

Solve the nominal and perturbation parts *separately*

Nominal (nonlinear) equation

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}_N(t), \mathbf{w}_N(t), t], \quad \mathbf{x}_N(0) \ given$$

Perturbation (linear) equation

$$\Delta\dot{\mathbf{x}}(t) \approx \mathbf{F}(t)\Delta\mathbf{x}(t) + \mathbf{G}(t)\Delta\mathbf{u}(t) + \mathbf{L}(t)\Delta\mathbf{w}(t), \quad \Delta\mathbf{x}(0) \ given$$

Approximate total solution

$$\mathbf{x}(t) \approx \mathbf{x}_N(t) + \Delta\mathbf{x}(t)$$

Jacobian Matrices Express Sensitivity to Small Perturbations Along Nominal Trajectory

$$\mathbf{F}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \quad ; \quad \mathbf{G}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \quad ; \quad \mathbf{L}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}}$$

Sensitivity to state perturbations: stability matrix

$$\mathbf{F}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} = \left[\begin{array}{cccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{array}\right]_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}}$$

Sensitivity to Small Control and Disturbance Perturbations Along Nominal Trajectory

$$\mathbf{F}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \quad ; \quad \mathbf{G}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} \quad ; \quad \mathbf{L}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}}$$

Control-effect matrix

$$\mathbf{G}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial u_1} & \ldots & \frac{\partial f_1}{\partial u_m} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial u_1} & \ldots & \frac{\partial f_n}{\partial u_m} \end{array}\right]_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}}$$

Disturbance-effect matrix

$$\mathbf{L}(t) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial w_1} & \ldots & \frac{\partial f_1}{\partial w_s} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial w_1} & \ldots & \frac{\partial f_n}{\partial w_s} \end{array}\right]_{\substack{\mathbf{x}=\mathbf{x}_N(t)\\ \mathbf{u}=\mathbf{u}_N(t)\\ \mathbf{w}=\mathbf{w}_N(t)}}$$

Stiffening Cubic Spring Example

Nonlinear equation, no inputs

$$\dot{x}_1(t) = f_1 = x_2(t)$$
$$\dot{x}_2(t) = f_2 = -10x_1(t) - 10x_1^{\ 3}(t) - x_2(t)$$

Integrate nonlinear equation to produce nominal path

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \rightarrow \int_0^{t_f} \begin{bmatrix} f_{1_N} \\ f_{2_N} \end{bmatrix} dt \rightarrow \begin{bmatrix} x_{1_N}(t) \\ x_{2_N}(t) \end{bmatrix} \quad in \quad [0, t_f]$$

Stiffening Cubic Spring Example

Evaluate partial derivatives along the path

$$\mathbf{F}(t) = \left[\begin{array}{c|c} \dfrac{\partial f_1}{\partial x_1} = 0 & \dfrac{\partial f_1}{\partial x_2} = 1 \\ \hline \dfrac{\partial f_2}{\partial x_1} = -10 - 30x_{1_N}^{\ 2}(t) & \dfrac{\partial f_2}{\partial x_2} = -1 \end{array}\right]$$

$$\mathbf{G}(t) = \begin{bmatrix} \dfrac{\partial f_1}{\partial u} = 0 \\ \dfrac{\partial f_2}{\partial u} = 0 \end{bmatrix} \qquad \mathbf{L}(t) = \begin{bmatrix} \dfrac{\partial f_1}{\partial w} = 0 \\ \dfrac{\partial f_2}{\partial w} = 0 \end{bmatrix}$$

Nominal and Perturbation Dynamic Equations

Nominal system, NTI

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t)], \quad \mathbf{x}_N(0)\ given$$
$$\dot{x}_{1_N}(t) = x_{2_N}(t)$$
$$\dot{x}_{2_N}(t) = -10x_{1_N}(t) - 10x_{1_N}{}^3(t) - x_{2_N}(t)$$

$$\begin{bmatrix} x_{1_N}(0) \\ x_{2_N}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \end{bmatrix}$$

New initial condition:

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

Perturbation system, LTV

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\Delta\mathbf{x}(t), \quad \Delta\mathbf{x}(0)\ given$$

$$\begin{bmatrix} \Delta\dot{x}_1(t) \\ \Delta\dot{x}_2(t) \end{bmatrix} = \left[\begin{array}{c|c} 0 & 1 \\ \hline -\left(10 + 30x_{1_N}{}^2(t)\right) & -1 \end{array}\right] \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix}$$

$$\begin{bmatrix} \Delta x_1(0) \\ \Delta x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$x_{1_N}(t)$ generated by nominal solution

Nominal and Perturbation Dynamic Solutions for Cubic Spring Example with $\mathbf{x}_N(0) = 0$

If nominal solution remains at equilibrium

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t)], \quad \mathbf{x}_N(0) = 0, \quad \mathbf{x}_N(t) = 0 \text{ in } [0,\infty]$$

Linearization is time-invariant

$$\begin{bmatrix} \Delta\dot{x}_1(t) \\ \Delta\dot{x}_2(t) \end{bmatrix} = \left[\begin{array}{c|c} 0 & 1 \\ \hline -10 & -1 \end{array}\right] \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix}$$

= Linear, Time-Invariant (LTI) System

Comparison of Approximate and Exact Solutions

Rate Initial Condition:

$x_{2_N}(0) = 9$

$\Delta x_2(0) = 1$

$x_2(t) = 10$

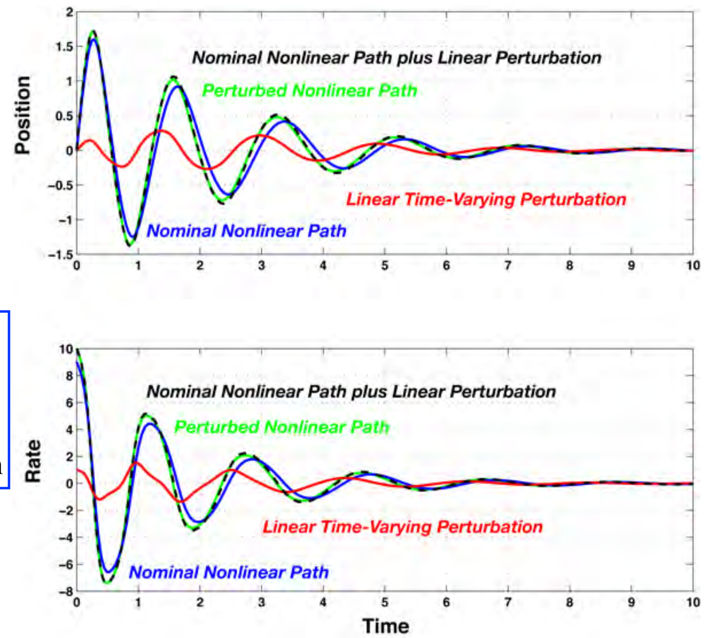
$x_{2_N}(t) + \Delta x_2(t) = 10$

$\mathbf{x}_N(t)$: Nominal

$\Delta\mathbf{x}(t)$: Perturbation

$\mathbf{x}(t)$: Actual

$\mathbf{x}_N(t) + \Delta\mathbf{x}(t)$: Approximation



Initial-Condition Response of a Linear, Time-Invariant (LTI) Model

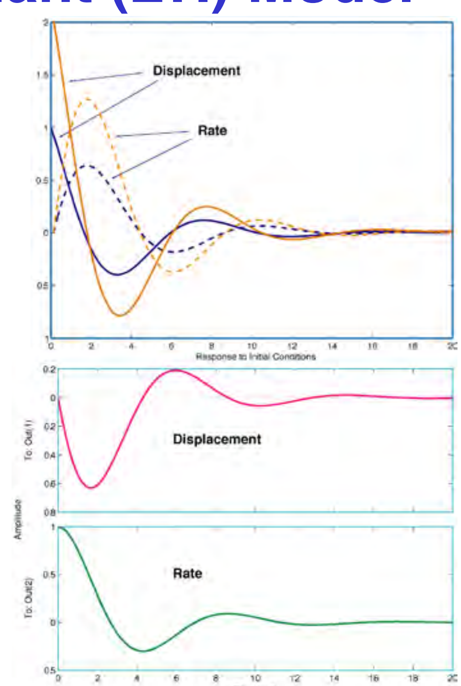
- **Doubling the initial condition doubles the output**
- **Stability, speed of response, and damping are independent of the initial condition**

```
%   Linear Model - Initial Condition
    F       =  [-0.5572 -0.7814;0.7814 0];
    G       =  [1 -1;0 2];
    Hx      =  [1 0;0 1];
    sys     =  ss(F, G, Hx,0);

    xo          =  [1;0];
    [y1,t1,x1]  =  initial(sys, xo);
    xo          =  [2;0];
    [y2,t2,x2]  =  initial(sys, xo);

    plot(t1,y1,t2,y2)

    figure
    xo      =  [0;1];
    initial(sys, xo)
```

$$\mathbf{F} = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$$

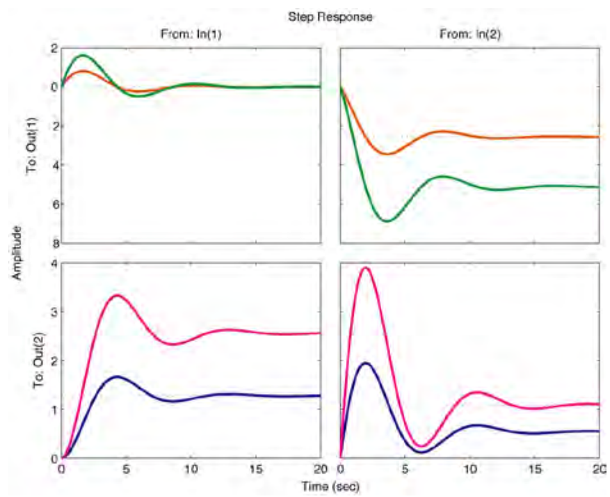
$$\mathbf{H}_x = \mathbf{I}_2; \quad \mathbf{H}_u = \mathbf{0}$$



Step Response of a Linear, Time-Invariant Model

- **Doubling the step input doubles the output**
- **Stability, speed of response, and damping are independent of the input**

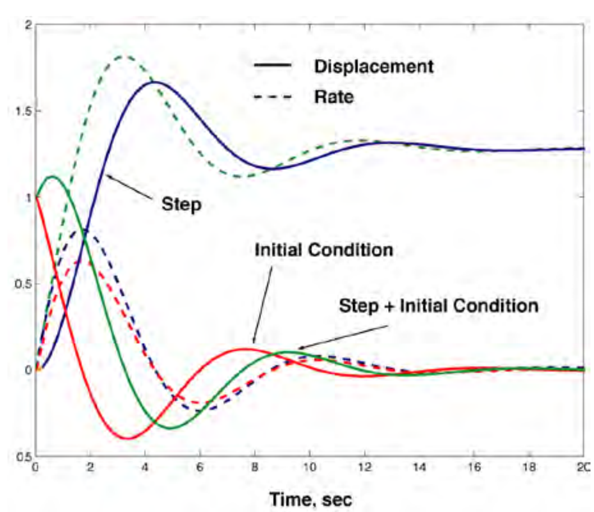
```
%   Linear Model - Step
    F   =  [-0.5572 -0.7814;0.7814 0];
    G   =  [1 -1;0 2];
    Hx   =  [1 0;0 1];
    sys =   ss(F, G, Hx,0);
    sys2    =    ss(F, 2*G, Hx,0);

%   Step response
    step(sys, sys2)
```



Response to Combined Initial Condition and Step Input

Linear system responses are additive

```
%  Linear Model - Superposition
   F      =  [-0.5572 -0.7814;0.7814 0];
   G      =  [1;0];
   Hx     =  [1 0;0 1];
   sys    =  ss(F, G, Hx,0);

   xo     =  [1; 0];
   t     =  [0:0.2:20];
   u      =  ones(1,length(t));

   [y1,t1,x1]  =  lsim(sys,u,t,xo);
   [y2,t2,x2]  =  lsim(sys,u,t);

   u          =  zeros(1,length(t));
   [y3,t3,x3]  =  lsim(sys,u,t,xo);

   plot(t1,y1,t2,y2,t3,y3)
```



Initial-Condition Responses of 1st-Order LTI Systems are Exponentials

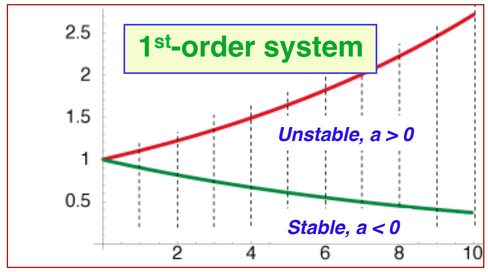
$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t), \quad \Delta\mathbf{x}(0)\ given$$

State vector is a scalar

$$\mathbf{F} = [a]$$

$$\Delta\dot{x}(t) = a\Delta x(t)$$

$$\Delta x(0)\ \text{given}$$



$$\Delta x(t) = \int_0^t \Delta\dot{x}(t)\,dt = \int_0^t a\Delta x(t)\,dt = e^{at}\Delta x(0)$$

Initial-Condition Responses of 2nd-Order LTI Systems

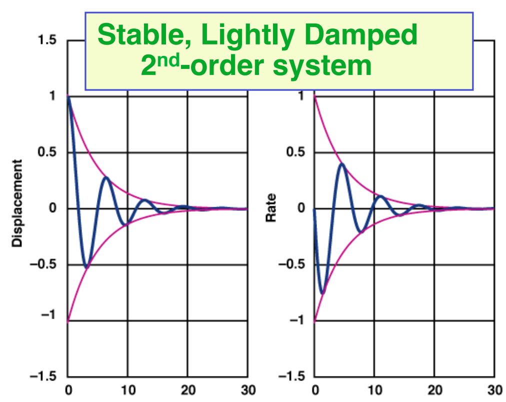
Exponentials and Sinusoids

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t)$$

$$\Delta\mathbf{x}(0)\ given$$

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

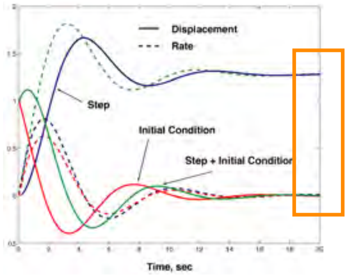
$$\Delta\dot{x}_1 = f_{11}\Delta x_1 + f_{12}\Delta x_2 + g_1\Delta u$$
$$\Delta\dot{x}_2 = f_{21}\Delta x_1 + f_{22}\Delta x_2 + g_2\Delta u$$



Sinusoid with exponential envelope

$$\Delta x_1(t) = A_1 e^{-\zeta\omega_n t}\cos\left[\omega_n\sqrt{1-\zeta^2}\,t + \varphi_1\right]$$

$$\Delta x_2(t) = A_2 e^{-\zeta\omega_n t}\cos\left[\omega_n\sqrt{1-\zeta^2}\,t + \varphi_2\right]$$



Equilibrium Response of Linear, Time-Invariant Models

- General equation

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t) + \mathbf{L}\Delta\mathbf{w}(t)$$

- At equilibrium,
  - Derivative goes to zero
  - State is unchanging

$$\mathbf{0} = \mathbf{F}\Delta\mathbf{x}(t) + \mathbf{G}\Delta\mathbf{u}(t) + \mathbf{L}\Delta\mathbf{w}(t)$$

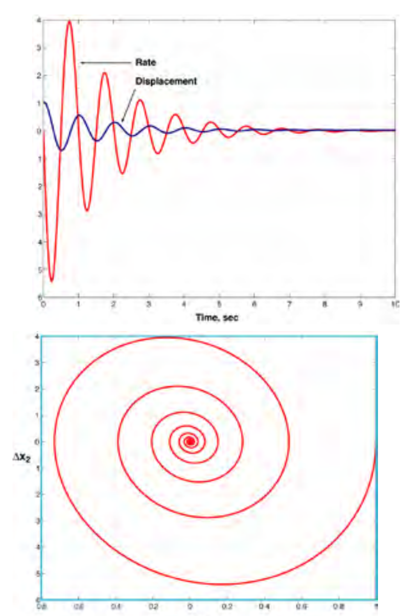
- State at equilibrium

$$\Delta\mathbf{x}^* = -\mathbf{F}^{-1}\left(\mathbf{G}\Delta\mathbf{u}^* + \mathbf{L}\Delta\mathbf{w}^*\right)$$
$$= -\frac{Adj(\mathbf{F})}{\det(\mathbf{F})}\left(\mathbf{G}\Delta\mathbf{u}^* + \mathbf{L}\Delta\mathbf{w}^*\right)$$

State ("Phase")-Plane Plots

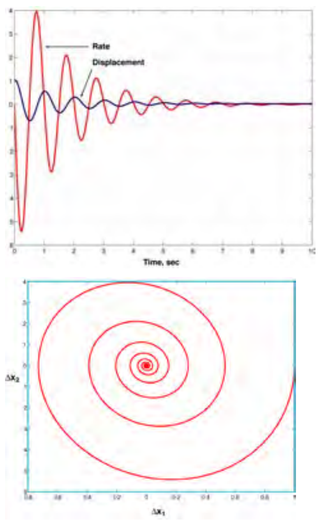
- Cross-plot of one component against another
- Time or frequency not shown explicitly in phase plane

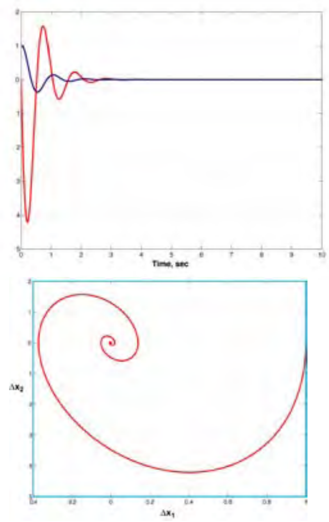
```
%   2nd-Order Model - Initial Condition Response
   clear
   z          =  0.1;  % Damping ratio
   wn         =  6.28; % Natural frequency, rad/s
   F          =  [0 1;-wn^2 -2*z*wn];
   G          =  [1 -1;0 2];
   Hx         =  [1 0;0 1];
   sys        =  ss(F, G, Hx,0);
   t          =  [0:0.01:10];
   xo         =  [1;0];
   [y1,t1,x1] =  initial(sys, xo, t);

   plot(t1,y1)
   grid on

   figure
   plot(y1(:,1),y1(:,2))
   grid on
```

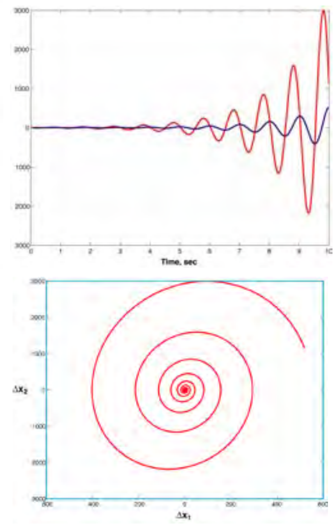


Effects of Damping Ratio on State-Plane Plots

- Damping ratio = 0.1
- Damping ratio = 0.3
- Damping ratio = –0.1

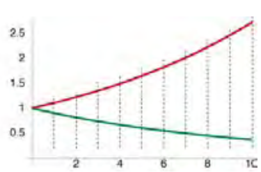




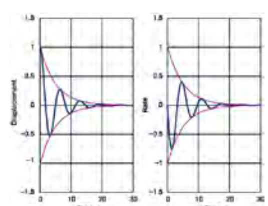


Response of Higher-Order LTI Systems

Superposition of Sub-System Responses



$$\mathbf{F}_{System} = \left[\begin{array}{c:c} \left[\mathbf{F}_{System\ 1}\right] & \left[\begin{array}{c}\text{Effect of}\\ \text{\#2 on \#1}\end{array}\right] \\ \hdashline \left[\begin{array}{c}\text{Effect of}\\ \text{\#1 on \#2}\end{array}\right] & \left[\mathbf{F}_{System\ 2}\right] \end{array}\right]$$



- Third-order system with uncoupled 1st- and 2nd-order sub-systems

$$\mathbf{F} = \left[\begin{array}{c|cc} f_{11} & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & f_{32} & f_{33} \end{array}\right]$$

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix}$$

- Coupling in first row and first column

$$\mathbf{F} = \left[\begin{array}{c|cc} f_{11} & f_{12} & f_{13} \\ \hline f_{21} & 0 & 1 \\ f_{31} & f_{32} & f_{33} \end{array}\right]$$

Examples of Coupled and Uncoupled Third-Order Systems

Third-order system with uncoupled 1st- and 2nd-order sub-systems

$$\mathbf{F} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -1.414 \end{bmatrix} \qquad \Delta\mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix}$$

Coupling in first row and first column

Position Coupling, Δx_3

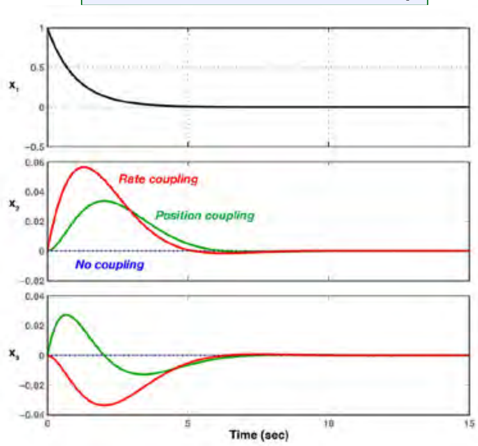
$$\mathbf{F} = \begin{bmatrix} -1 & 0.1 & 0 \\ 0.1 & 0 & 1 \\ 0 & -1 & -1.414 \end{bmatrix}$$

Rate Coupling, Δx_2

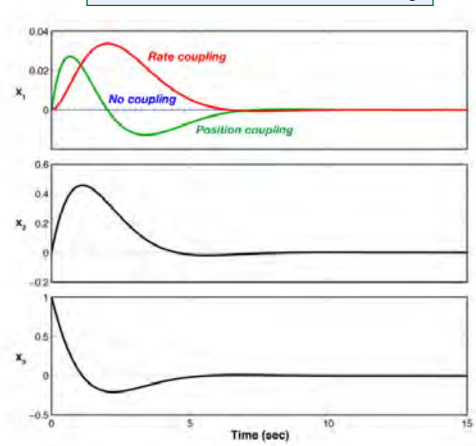
$$\mathbf{F} = \begin{bmatrix} -1 & 0 & 0.1 \\ 0 & 0 & 1 \\ 0.1 & -1 & -1.414 \end{bmatrix}$$

3rd-Order LTI Systems with Coupled Response

Initial Condition on Δx_1



Initial Condition on Δx_3



Next Time:
Dynamic Effects of Feedback Control

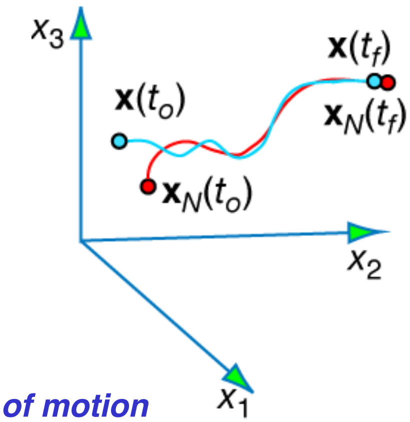
Supplemental Material

Perturbed Initial Conditions Produce Perturbed Path

- **Given**
  - Initial condition, control, and disturbance histories

$$\mathbf{x}(t_0), \mathbf{u}(t), \mathbf{w}(t) \quad \text{for } t \text{ in } [t_o, t_f]$$

  - Path (or ***trajectory***) is approximated by executing a numerical algorithm
  - Perturbing the initial condition produces a new path



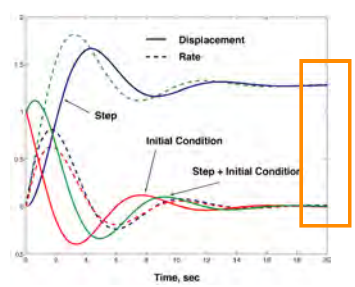
- **Both paths satisfy the *same equations of motion***

$$\dot{\mathbf{x}}_N(t) = \mathbf{f}[\mathbf{x}_N(t), \mathbf{u}(t), \mathbf{w}(t), t], \quad \mathbf{x}_N(t_0) \quad given$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t], \quad \mathbf{x}(t_0) \quad given$$

- $\mathbf{x}_N$: Nominal path
- $\mathbf{x}$: Perturbed path

Equilibrium Response of Second-Order LTI System



System description

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}; \quad \mathbf{G} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}; \quad \mathbf{L} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$$

State equilibrium depends on constant input values

$$\begin{bmatrix} \Delta x_1 * \\ \Delta x_2 * \end{bmatrix} = -\frac{\begin{bmatrix} f_{22} & -f_{12} \\ -f_{21} & f_{11} \end{bmatrix}}{(f_{11}f_{22} - f_{12}f_{21})} \left[\begin{pmatrix} g_1 \\ g_2 \end{pmatrix} \Delta u * + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \Delta w * \right]$$

$$|\mathbf{F}| = (f_{11}f_{22} - f_{12}f_{21}) \neq 0$$