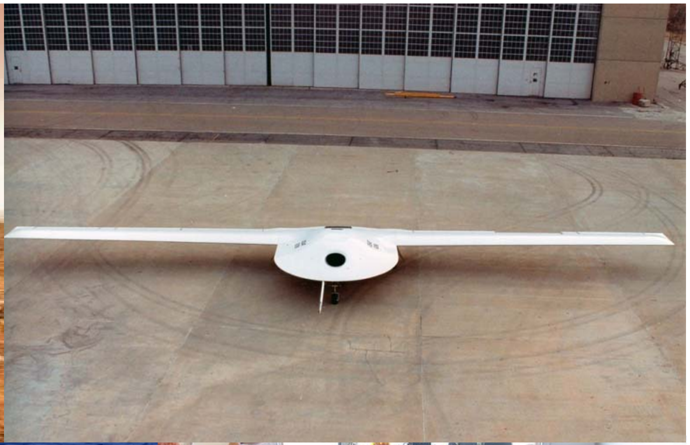





ME597: Autonomous Mobile Robotics
Section 2 – Linear Systems

Prof. Steven Waslander

LINEAR ALGEBRA

- Scalar, Vector, Matrix

$$c \in \mathbb{R} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n \qquad A = \begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & \ddots & \\ \vdots & & A_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

  - Fat matrix: n<m, Skinny matrix: n>m

- Unit Vector, Identity Matrix

$$e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} \qquad I = \begin{bmatrix} 1 & 0 & \cdots \\ 0 & 1 & \\ \vdots & & \ddots \end{bmatrix}$$

Linear Algebra

- Matrix Transpose

$$A^T = \begin{bmatrix} A_{11} & A_{21} & \cdots \\ A_{12} & \ddots & \\ \vdots & & \end{bmatrix}$$

- Matrix Addition

$$A+B = \begin{bmatrix} A_{11}+B_{11} & A_{12}+B_{12} & \cdots \\ A_{21}+B_{21} & \ddots & \\ \vdots & & \end{bmatrix}$$

- Matrix Multiplication

$$AB = \begin{bmatrix} \sum_i A_{1i}B_{i1} & \sum_i A_{1i}B_{i2} & \cdots \\ \sum_i A_{2i}B_{i1} & \ddots & \\ \vdots & & \end{bmatrix}$$

Linear Algebra

- Matrix Transpose of Added Matrices

$$(A+B)^T = A^T + B^T$$

- Matrix Transpose of Multiplied Matrices

$$(AB)^T = B^T A^T$$

- Quadratic form

$$\begin{aligned}(Ax+b)^T(Ax+b) &= x^T A^T Ax + x^T A^T b + b^T Ax + b^T b \\ &= x^T A^T Ax + 2x^T A^T b + b^T b \\ &= \underbrace{x^T Cx}_{\text{Quadratic term}} + d^T x + e\end{aligned}$$

Linear Algebra

- Matrix Rank: $\rho(A)$
  - The number of independent rows or columns
  - Nonsingular = Full Rank

$$\rho(A) = \min(n, m)$$

  - Singular = Not full rank

$$\rho(A) < \min(n, m)$$

    - Non-empty nullspace

$$\exists\ x \text{ such that } Ax = 0$$

- Matrix Inverse (square A)

$$AA^{-1} = A^{-1}A = I$$

  - Nonsingular and square <=> Invertible

Linear Algebra

- Matrix Trace

$$tr(A) = \sum_i A_{ii}$$

- Symmetric Matrix

$$A = A^T = \begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{12} & \ddots & \\ \vdots & & \end{bmatrix}$$

- Positive Definiteness (Semi-Definiteness)
  - For a symmetric nXn matrix A, and for any x in R^n

$$x^T A x > 0 \qquad (x^T A x \geq 0)$$

Linear Algebra

- Eigenvalues and Eigenvectors of a matrix
  - For a matrix A, the vector x is an *eigenvector* of A with a corresponding *eigenvalue* λ if they satisfy the equation

$$Ax = \lambda x$$

- The eigenvalues of a diagonal matrix are its diagonal elements
- The inverse of A exists if and only if (iff) none of the eigenvalues are zero
- Positive definite A has all eigenvalues greater than zero

Linear Algebra

- Differentiation of linear matrix equation

$$\frac{d}{dx}\left(Ax\right) = A$$

$$\frac{d}{dx}\left(x^T A\right) = A^T$$

- Differentiation of a quadratic matrix equation

$$\frac{d}{dx}\left(x^T Ax\right) = x^T A + x^T A^T$$

Linear Algebra

- Least Squares Solution
  - If A is a skinny matrix (n>m), and we wish to find x for which

$$Ax = b$$

  - Since A is skinny, the problem is over-constrained
    - No solution exists
  - Instead, minimize the square of the error between Ax and b

$$\min_x \| Ax - b \|_2^2$$

$$= \min_x (Ax - b)^T (Ax - b)$$

$$= \min_x x^T A^T Ax - 2b^T Ax + b^T b$$

Linear Algebra

- Setting the derivative to zero

$$2x^T A^T A - 2b^T A = 0$$

$$A^T A x = A^T b$$

$$x = \left(A^T A\right)^{-1} A^T b$$

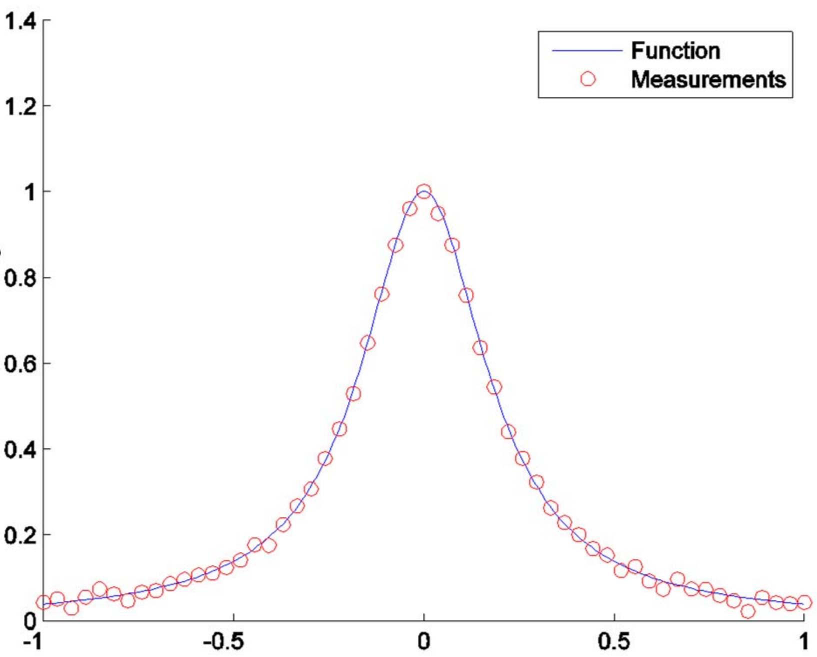
$$x = A^{\dagger} b$$

  - Known as the pseudo-inverse

  - This methodology is used over and over in the course
    - Quadratic cost minimized to find closed form solution

Linear Algebra

- Least Squares example
  - Data fitting with polynomials
    - Given a function of interest



$$g(t) = \frac{1}{1 + 25t^2}$$

    - And a set of measurements $b(t_m)$of that function at points t_m

$$b(t_m), \quad t_m = \{-1, -0.99, \ldots, 1\}$$

    - Find the best polynomial fit for polynomial $f_P(t)$ of order P

$$f_P(t) = x_1 + x_2 t + \ldots + x_{P+1} t^P$$

Linear Algebra

- Least Squares example
  - Can formulate this as a least squares problem where we want to minimize the mean square error between polynomial prediction and measurement at each t_m:

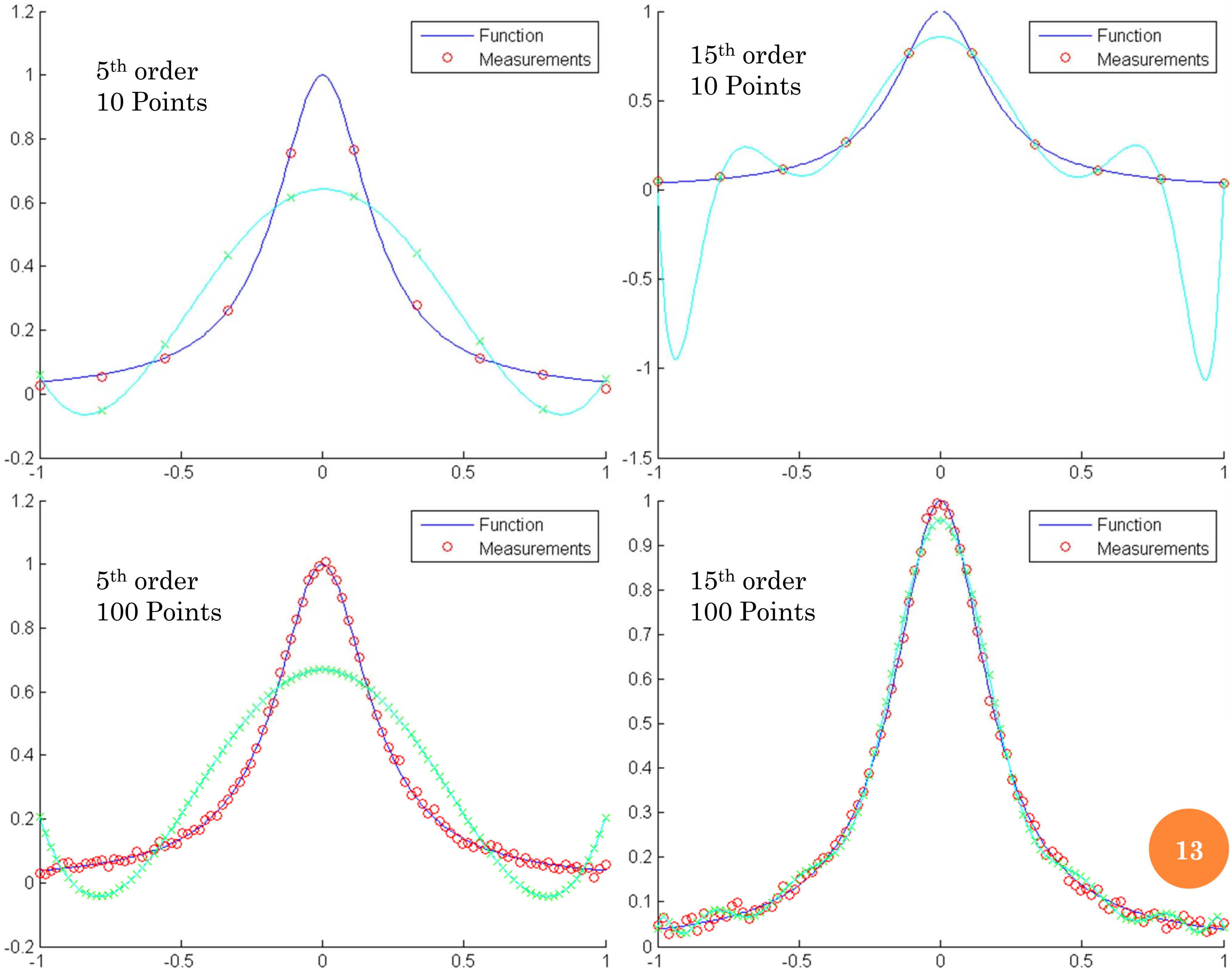
$$\min_x \| f_P(t_m) - b(t_m) \|_2^2$$

  - The polynomial can be written as

$$f_P(t_m) = A(t_m)x = x_1 + x_2 t_m + \ldots + x_{P+1} t_m{}^P$$

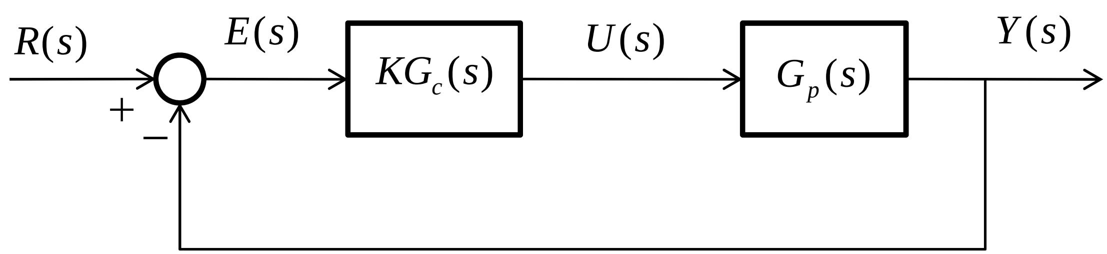
$$A(t_m) = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^P \\ 1 & t_2 & t_2^2 & \cdots & t_2^P \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^P \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{bmatrix}$$

  - Use least squares solution method to find coefficients of f.


5th order
10 Points
Function
Measurements
15th order
10 Points
Function
Measurements
5th order
100 Points
Function
Measurements
15th order
100 Points
Function
Measurements
-1
-0.5
0
0.5
1

SISO CONTROL

- One input, one output, one transfer function between the two



Closed loop transfer function

$$\frac{Y(s)}{R(s)} = T(s)$$

- Model requires
  - transfer function from single input to single output
  - initial conditions to start from (usually assumed 0)
- Model hides inner workings of plant

State Space: A new system model

- Multi-Input-Multi-Output (MIMO) model, maintains complete plant picture
  - Matrix and vector notation, use power of linear algebra for many key results
- Definition: *The state of a system is a vector of system variables that entirely defines the system at a specific instance in time.*
  - Example: at t=0, initial conditions define a state vector.
  - Entire history of state variables can be discarded, only need current state and system dynamics to continue forward in time.

Standard Form Dynamics

- Linear first order time-invariant dynamics in continuous time
  - Update equation

$$\dot{x}(t) = Ax(t) + Bu(t)$$

  - Measurement equation

$$y(t) = Cx(t) + Du(t)$$

  - A, B matrices: state derivatives can depend on any state or input variable
  - C, D matrices: output can depend on any state or input variable

Standard Form Dynamics

- Linear first order time-invariant dynamics in discrete time
  - Update equation, timesteps indexed by t

$$x_t = Ax_{t-1} + Bu_t$$

  - Measurement equation

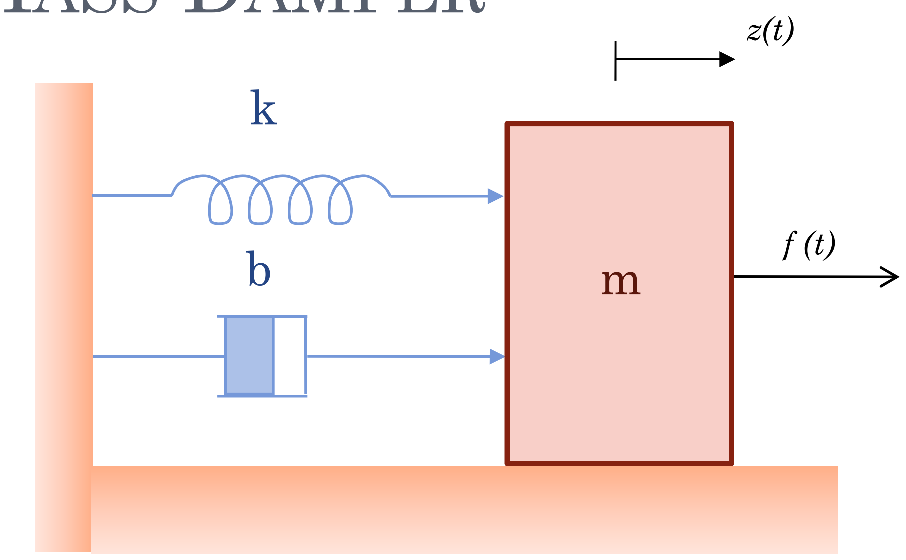
$$y_t = Cx_t + Du_t$$

  - A, B matrices: state update can depend on any state or input variable
  - C, D matrices: measurements can depend on any state or input variable

EXAMPLE – SPRING MASS DAMPER

- Equation of Motion

$$m\ddot{z}(t) + b\dot{z}(t) + kz(t) = f(t)$$



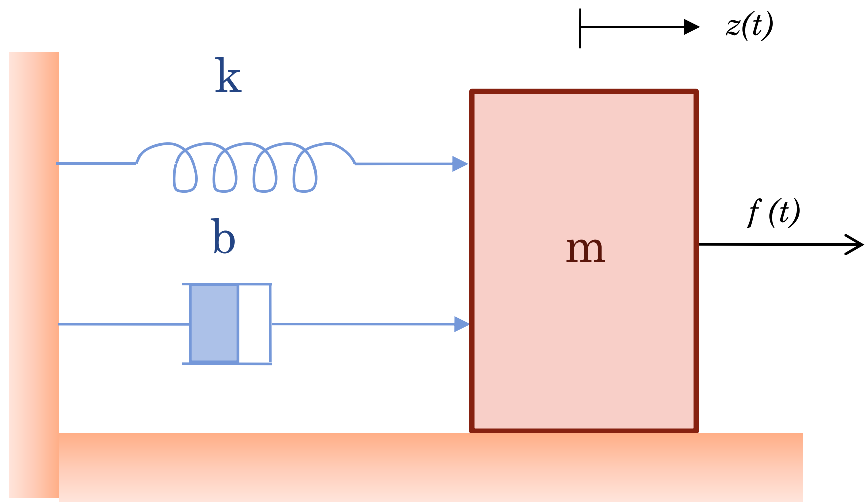
- Transfer function

$$\frac{Z(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

- Four variables in ODE: one input, one variable defined by ODE, two states remain.

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}$$

Example Cont'd



- Motion Model

$$\begin{bmatrix} \dot{z}(t) \\ \ddot{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-b}{m} \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t)$$

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Measurement model: position and velocity sensors

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} f(t)$$

$$y = Cx(t) + Du(t)$$

In Transfer Function Form

- Take Laplace transform of update equation

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \quad \Rightarrow \quad sX(s) - x(0) = AX(s) + BU(s)$$

- Solve for X(s), with x(0)=0

$$(sI - A)X(s) = BU(s)$$

$$X(s) = (sI - A)^{-1}BU(s)$$

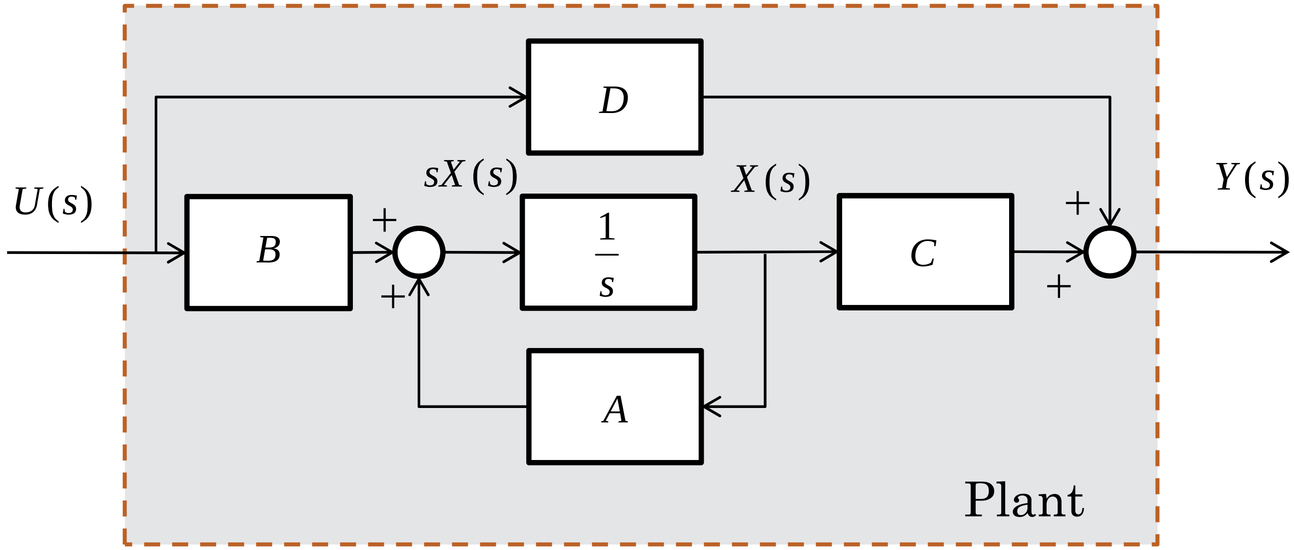
- Combine with measurement model

$$Y(s) = CX(s) + DU(s)$$

$$\frac{Y(s)}{U(s)} = C\left(sI - A\right)^{-1}B + D$$

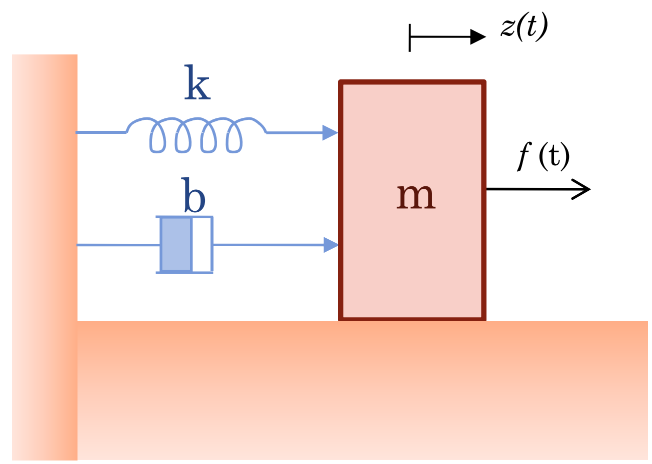
BLOCK DIAGRAM

- Continuous LTI State space model as a block diagram (Laplace Domain)



EXAMPLE: FIND TFS

$$A = \begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1} B$$

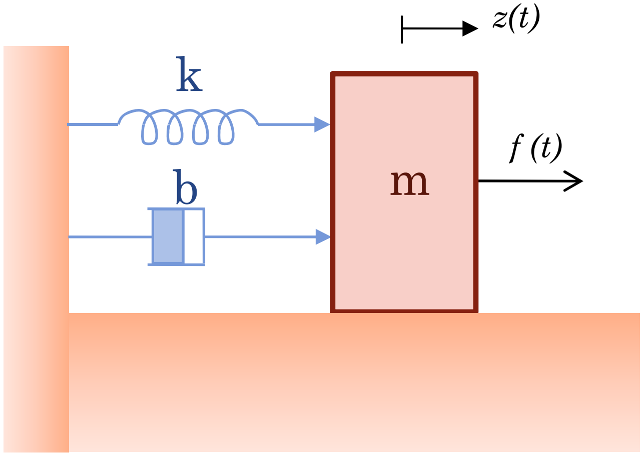
$$\frac{Y(s)}{U(s)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{b}{m} \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix}$$

$$\frac{Y(s)}{U(s)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} s & -1 \\ \dfrac{k}{m} & s + \dfrac{b}{m} \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix}$$

Example: Find TFs

$$\frac{Y(s)}{U(s)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{\det(sI - A)} \left(\begin{bmatrix} s + \frac{b}{m} & 1 \\ \frac{-k}{m} & s \end{bmatrix} \right) \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$

$$\frac{Y(s)}{U(s)} = \begin{bmatrix} \frac{1}{\det(sI - A)} \\ \frac{s}{\det(sI - A)} \end{bmatrix} = \begin{bmatrix} \frac{\frac{1}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} \\ \frac{\frac{s}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} \end{bmatrix}$$



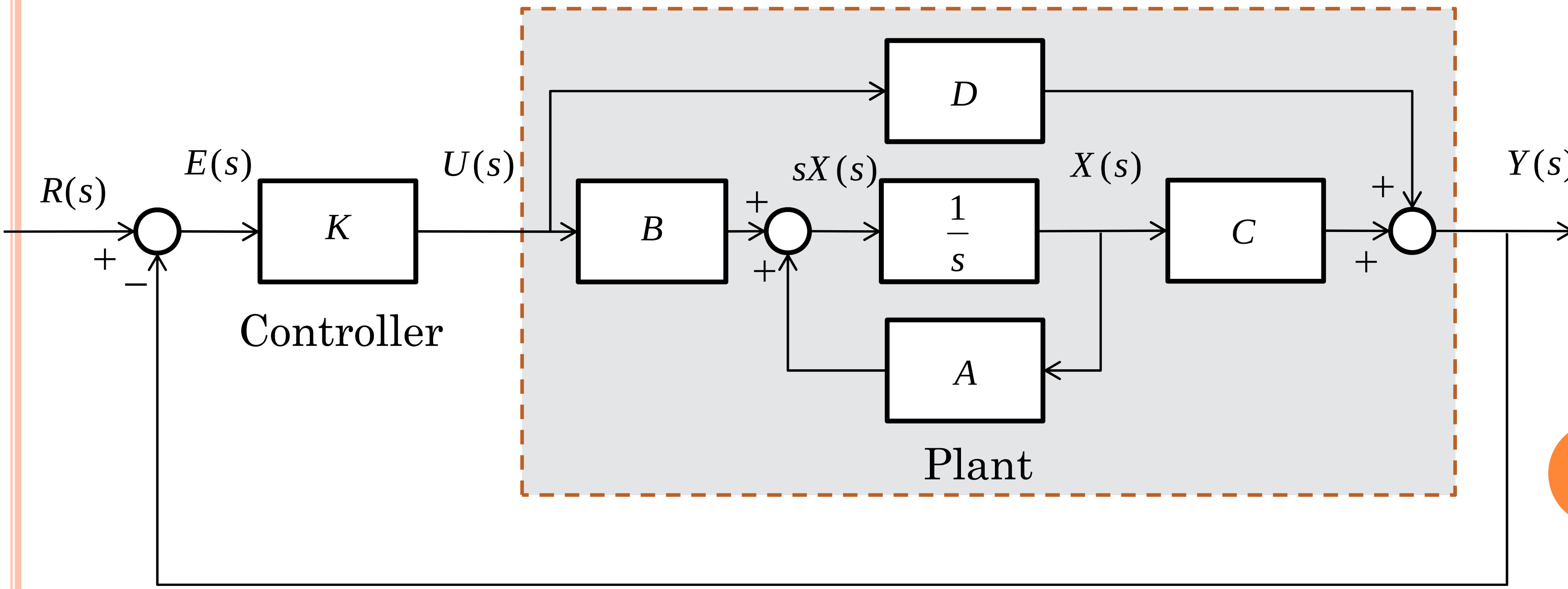
- So roots of det*(sI-A)* determine poles of open loop system
  - Well known equation in linear algebra: eigenvalues/eigenvectors
  - Open loop poles are eigenvalues of A
  - Holds for all sizes of A, not just 2X2

State feedback control

- If $C = I$ and $D = 0$, full state feedback
  - More than one signal, in fact everything we could possibly need

$$u(t) = -Kx(t)$$

  - Assume $R(s) = 0$ (regulator)



State Feedback

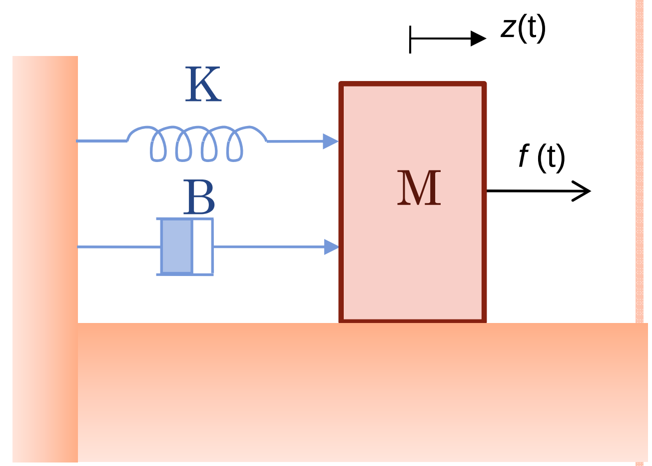
- Closed loop transfer function

$$\frac{X(s)}{R(s)} = (sI - A + BK)^{-1}$$

- Now, eigenvalues of A-BK are poles of closed loop system
- In fact, since there is one K for every eigenvalue, we can place the closed loop poles anywhere we'd like.

EXAMPLE: POLE PLACEMENT

- Lets place closed loop poles at $s = -5 \pm 5j$
- Note: $$BK = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{K_1}{m} & \frac{K_2}{m} \end{bmatrix}$$
- Match coefficients of polynomials
  - Desired = actual



$$s^2 + 10s + 50 = \det\left(\begin{bmatrix} s & -1 \\ \frac{k}{m} + \frac{K_1}{m} & s + \frac{b}{m} + \frac{K_2}{m} \end{bmatrix} \right)$$

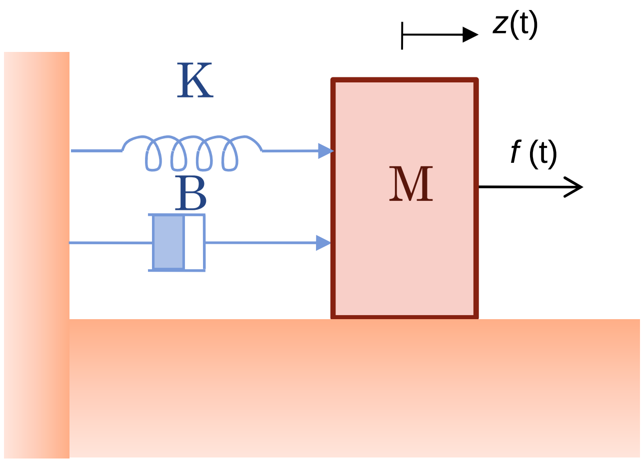
$$= s^2 + \left(\frac{K_2}{m} + \frac{b}{m} \right) s + \left(\frac{K_1}{m} + \frac{k}{m} \right)$$

Example: Controller

- What is full state feedback?

$$u(t) = K_1 z(t) + K_2 \dot{z}(t)$$

  - PD control
- To add integral control, add an integrator state



$$x(t) = \begin{bmatrix} \int z dt \\ z \\ \dot{z} \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix}$$

Controllability

- Can I get there from here?
  - A system is controllable if for any set of initial and final states, $x(0)$ and $x(T)$, there exists a control input sequence, $u(0)$ to $u(T)$, to get from $x(0)$ to $x(T)$.
  - Can be checked easily: The following matrix must be full rank

$$\text{rank}\left(\begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}\right) = n$$

OBSERVABILITY

- Can I see there from here?
  - Given any sequence of states $x(0)$ to $x(T)$, inputs $u(0)$ to $u(T)$ and outputs $y(0)$ to $y(T)$, a system is observable if the state can be uniquely determined from the outputs alone.

- Again, an easy check on the observability matrix determines if a system is observable

$$\text{rank}\left(\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) = n$$

Optimal Control

- Since we can place poles anywhere, can change objective of control design
- Minimize quadratic errors in states and quadratic use of inputs

$$\min_{x,u} \int_0^t x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)d\tau$$

  - Penalize big deviations more heavily that small ones
  - Quadratic cost and linear dynamics result in a time invariant control law, another way to set the gains for state feedback control

Optimal Estimation

- The second half of the model describes the relationship between the state and the measured outputs.
  - Any sensor dynamics must be included in the state
- In reality, both disturbances and noise will exist

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$

$$y(t) = Cx(t) + Du(t) + v(t)$$

- Assume w, v are Gaussian white noise with covariance Q, R
- Assume u(t) is known exactly
- Formulate minimum mean squared error estimation problem, results in Kalman filter

State Space Model