EECS 16B Midterm 2 Review Session

Presented by <NAMES >(HKN)

Disclaimer

This is an unofficial review session and HKN is not affiliated with this course. Although some of the presenters may be course staff, the material covered in the review session may not be an accurate representation of the topics covered in and difficulty of the exam.

If you want to follow along: head to <Slides location and/or link >

Slides are also posted at @# on Piazza.

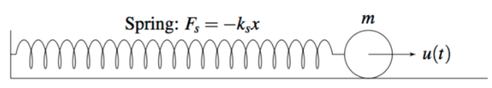
HKN Drop-In Tutoring

- HKN has office hours every weekday from **11 AM - 3 PM** and **8 PM - 10 PM** on hkn.mu/ohqueue
- The schedule of tutors can be found at hkn.mu/tutor
  <Itemize the presenter hours here >

State-Space Representations

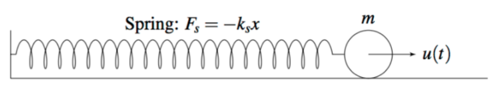
State Space Modeling: Example

Assume we have the following spring system:



State Space Modeling: Example

Assume we have the following spring system:



We can model the system as a linear continuous time state space model:

$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$

$\vec{y}(t) = C\vec{x}(t)$

in which:

$\vec{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}$, $A = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & 0 \end{bmatrix}$, $\vec{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$

State Space Modeling:

$$\begin{tikzcd}
\text{original space:} & \vec{x}(t) \arrow[r, "A"] \arrow[d, bend right, "T^{-1}"'] & \frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) \arrow[d, bend right, "T^{-1}"'] \\
\text{new space:} & \tilde{\vec{x}}(t) \arrow[r, "\tilde{A}"] \arrow[u, bend right, "T"'] & \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\vec{x}}(t) \arrow[u, bend right, "T"']
\end{tikzcd}$$

Always apply operations to right side first

$$\tilde{A} = T^{-1}AT$$

$$\tilde{\vec{x}}(t) = T^{-1}\vec{x}(t)$$

State Space Modeling Procedure:

1. Set up differential equation of the form:
   $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$

State Space Modeling Procedure:

1. Set up differential equation of the form:
   $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$

2. Find λ_i of A; let $\tilde{A} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$

State Space Modeling Procedure:

1. Set up differential equation of the form:
   $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
2. Find λ_i of A; let $\tilde{A} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$
3. Find eigenvectors $\vec{v_i}$ of A; let $T = \begin{bmatrix} \vec{v_1} & \vec{v_2} & \cdots & \vec{v_n} \end{bmatrix}$

State Space Modeling Procedure:

1. Set up differential equation of the form: $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
2. Find λ_i of A; let $\tilde{A} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$
3. Find eigenvectors $\vec{v_i}$ of A; let $T = \begin{bmatrix} \vec{v_1} & \vec{v_2} & \cdots & \vec{v_n} \end{bmatrix}$
4. Convert $\vec{x}(t)$ to $\tilde{\vec{x}}(t)$ using: $\tilde{\vec{x}}(t) = T^{-1}\vec{x}(t)$

State Space Modeling Procedure:

1. Set up differential equation of the form: $\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
2. Find λ_i of A; let $\tilde{A} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$
3. Find eigenvectors $\vec{v_i}$ of A; let $T = \begin{bmatrix} \vec{v_1} & \vec{v_2} & \cdots & \vec{v_n} \end{bmatrix}$
4. Convert $\vec{x}(t)$ to $\tilde{\vec{x}}(t)$ using: $\tilde{\vec{x}}(t) = T^{-1}\vec{x}(t)$
5. Solve $\frac{d}{dt}\tilde{\vec{x}}(t) = \tilde{A}\tilde{\vec{x}}(t) + T\vec{b}u(t)$

State Space Modeling Procedure:

1. Set up differential equation of the form:
$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = A\vec{x}(t) + \vec{b}u(t)$
2. Find λ_i of A; let $\tilde{A} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$
3. Find eigenvectors $\vec{v_i}$ of A; let $T = \begin{bmatrix} \vec{v_1} & \vec{v_2} & \cdots & \vec{v_n} \end{bmatrix}$
4. Convert $\vec{x}(t)$ to $\tilde{\vec{x}}(t)$ using: $\tilde{\vec{x}}(t) = T^{-1}\vec{x}(t)$
5. Solve $\frac{d}{dt}\tilde{\vec{x}}(t) = \tilde{A}\tilde{\vec{x}}(t) + T\vec{b}u(t)$
6. Convert solution back to $\vec{x}(t)$

State Space Modeling:

Continuous time solution:

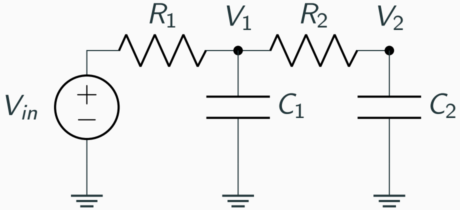
$$\tilde{x}(t) = e^{\lambda t}\tilde{x}(0) + \frac{e^{\lambda t} - 1}{\lambda}u(t) + w(t)$$

Discrete time solution:

$$x_d(i+1) = e^{\lambda \Delta}x_d(i) + \frac{e^{\lambda \Delta} - 1}{\lambda}u(i) + w(i)$$

State Space Modeling Example:

Given the following circuit:



in which $R_1 = 2\,\Omega$, $R_2 = \frac{8}{3}\Omega$, $C_1 = 1\,\mathrm{C}$, $C_2 = \frac{3}{2}\mathrm{C}$
solve equations for V_1 and V_2

Stability, Observability, and Controllability

Stability, Observability, Controllability:

given:

$\vec{x}(i+1) = A\vec{x}(i) + Bu(i)$

$\vec{y}(i) = C\vec{x}(i)$

in which:

$\vec{x}$ is our state,

$\vec{u}$ is our input,

$\vec{y}$ is what we can observe:

Stability (Discrete time):

Discrete time model:
if $|\lambda_i| < 1$ for all λ_i of A, system is stable
intuition: if any $|\lambda_i| >= 1$, state vector is increasing each time step will be infinitely magnified over time

Stability (Continuous time):

Continuous time model:
if the real parts of all eigenvalues of A are strictly negative, system is stable
intuition: if real part of eigenvalue is positive, state vector is increasing over time and will be infinitely magnified over time

Controllability:

if $\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$ spans R^n, then system is controllable

Feedback:

if system is controllable, we can set: $u(t) = K\vec{x}(t)$
plugging in, we get: $\vec{x}(t+1) = (A + BK)\vec{x}(t)$
we can find the eigenvalues of (A + BK) to check for stability

Observability:

if $\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$ spans R^n, system is observable

intuition: if observability matrix is full rank, it is invertible, and we can retrieve all the past states without loss of information

Stability, Controllability, Observability Example:

given the following system:

$$\vec{x}[t+1] = \begin{bmatrix} -5 & 0 \\ 7 & 6 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u[t]$$

$$\vec{y}[t] = \begin{bmatrix} 1 & 1 \end{bmatrix} \vec{x}[t]$$

Stability Check:

$\lambda = 6, -5$

Stability Check:

$\lambda = 6, -5$

System is unstable

Controllability Check:

$$\begin{bmatrix} AB & B \end{bmatrix} = \begin{bmatrix} -10 & 2 \\ 8 & -1 \end{bmatrix} \text{ which spans } R^n$$

Controllability Check:

$$\begin{bmatrix} AB & B \end{bmatrix} = \begin{bmatrix} -10 & 2 \\ 8 & -1 \end{bmatrix} \text{ which spans } R^n$$

System is controllable

Observability Check:

$$\begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 6 \end{bmatrix} \text{ which spans } R^n$$

Observability Check:

$$\begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 6 \end{bmatrix} \text{ which spans } R^n$$

System is observable

Eigenvalue Placement

Eigenvalue Placement

Why?

- Recall that we are always interested in determining if a given system is BIBO (bounded input bounded output) stable.
- More precisely, if we have a system described by $\vec{x}(t+1) = A\vec{x}(t) + Bu(t) + \vec{\omega}(t)$ we would like the eigenvalues of $A \in \mathbb{R}^{n\times n}$, to satisfy the following property : $|\lambda_i| < 1$.
- So what if we have a λ that does not satisfy this property?
- This is where eigenvalue placement comes into play!
- Assuming the system is controllable, we will use closed loop controls to change the eigenvalues such that they satisfy this property.

How?

- Assume e.g. a DT system. Input: $u[t]$ If the system is controllable then we can use feedback, which means that we can let the input depend on the output, $\vec{x}[t]$.
- We would like to change the matrix multiplying $\vec{x}[t]$ such that $|\lambda_i| < 1$, so let's see what happens when we let $u[t] = K\vec{x}[t]$, where $K \in \mathbb{R}^{1\times n}$.
- Using this input we have:

$$\begin{aligned}\vec{x}[t+1] &= A\vec{x}[t] + Bu[t] + \vec{\omega}[t] \\ &= A\vec{x}[t] + BK\vec{x}[t] + \vec{\omega}[t] \\ &= (A+BK)\vec{x}[t] + \vec{\omega}[t]\end{aligned}$$

- Strategically choosing K allows us to have specific λ's for A + BK (Good!).
- This process is called coefficient matching.

Example

- Suppose we are given a controllable system defined by:

$$\vec{x}[t+1] = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u[t]$$

- Is the system stable?

Example

- Suppose we are given a controllable system defined by:

$$\vec{x}[t+1] = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u[t]$$

- Is the system stable? No! $\lambda = 2, 1$
- What if we let

$$u[t] = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

  Then we have:

$$\vec{x}(t+1) = \begin{bmatrix} -2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

- Solve for the values of f_1 and f_2 such that $\lambda_1 = -0.25$ and $\lambda_2 = 0$

Example

- Suppose we are given a controllable system defined by:

$$\vec{x}[t+1] = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u[t]$$

- Is the system stable? No! $\lambda = 2, 1$
- What if we let

$$u[t] = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

  Then we have:

$$\vec{x}(t+1) = \begin{bmatrix} -2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

- Solve for the values of f_1 and f_2 such that $\lambda_1 = -0.25$ and $\lambda_2 = 0$
- The answer is $f_1 = -1.50$ and $f_2 = 0.25$

Example

- Suppose we are given a controllable system defined by:

$$\vec{x}[t+1] = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u[t]$$

- Is the system stable? No! $\lambda = 2, 1$
- What if we let

$$u[t] = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

  Then we have:

$$\vec{x}(t+1) = \begin{bmatrix} -2 & -1 \\ 0 & 1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 2 \\ -1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 \end{bmatrix} \vec{x}[t]$$

- Solve for the values of f_1 and f_2 such that $\lambda_1 = -0.25$ and $\lambda_2 = 0$
- The answer is $f_1 = -1.50$ and $f_2 = 0.25$
- Although the process is very messy hopefully you see why eigenvalue placement is very important to stabilize systems.

Controllable Canonical Form

- Controllable Canonical Form (CCF) for any controllable system is a special form that allows us to simplify the process of eigenvalue placement. It takes on the following form:

$$A^* = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 1 \\ \alpha_0 & \alpha_1 & \dots & \dots & \alpha_{n-1} \end{bmatrix} \qquad B^* = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Controllable Canonical Form

- Controllable Canonical Form (CCF) for any controllable system is a special form that allows us to simplify the process of eigenvalue placement. It takes on the following form:

$$A^* = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 1 \\ \alpha_0 & \alpha_1 & \dots & \dots & \alpha_{n-1} \end{bmatrix} \qquad B^* = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

- The characteristic polynomial of A^* is $\lambda_n - \sum_{i=0}^{n-1} \alpha_i \lambda^i = 0$.
- So how does it help with eigenvalue placement?

Controllable Canonical Form

- Controllable Canonical Form (CCF) for any controllable system is a special form that allows us to simplify the process of eigenvalue placement. It takes on the following form:

$$A^* = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & 1 \\ \alpha_0 & \alpha_1 & \dots & \dots & \alpha_{n-1} \end{bmatrix} \qquad B^* = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

- The characteristic polynomial of A^* is $\lambda_n - \sum_{i=0}^{n-1} \alpha_i \lambda^i = 0$.
- So how does it help with eigenvalue placement? The last row of this matrix determines the eigenvalues of A^* so modifying the last row will allow us to (easily) modify the eigenvalues.

How to convert to CCF

- Let A, B be the matrices in standard form and let A^*, B^* be the matrices in CCF.

How to convert to CCF

- Let A, B be the matrices in standard form and let A^*, B^* be the matrices in CCF.
- Recall the matrix we used to check controllability?

How to convert to CCF

- Let A, B be the matrices in standard form and let A^*, B^* be the matrices in CCF.
- Recall the matrix we used to check controllability?

$$C = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$$

$$C^* = \begin{bmatrix} B^* & A^*B^* & \dots & A^{*n-1}B^* \end{bmatrix}$$

- We then have $T := C^*C^{-1}$, such that $A^* = TAT^{-1}$ and $B^* = TB$.
- Remember, all controllable matrices with single input can be transformed into CCF!

Example

Consider the following discrete time system:

$$\vec{x}[t+1] = \begin{bmatrix} -2 & 2 & 0 \\ 2 & -4 & 2 \\ 0 & 1 & -1 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u[t]$$

(a) Is the system stable? Is it controllable?

(b) Using an appropriate transformation ($\vec{z}[t] = T\vec{x}[t]$), bring the system to controllable canonical form.

(c) Using the state feedback $u[t] =$

$$\begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}$$

$\vec{z}[t]$ bring the eigenvalues of the system to $0, 0.75, -0.25$.

Solutions to Example

(a) The characteristic polynomial is:
$\lambda^3 + 7\lambda^2 + 8\lambda = \lambda(\lambda^2 + 7\lambda + 8) = 0$, therefore the eigenvalues of A are $\{0, -5.56, -1.44\}$. As we can see there are $|\lambda_i| > 1$ therefore the system is not stable.
The controllability matrix $C =$

$$\begin{bmatrix} 1 & -2 & 8 \\ 0 & 2 & -12 \\ 0 & 0 & 2 \end{bmatrix}$$

C has full rank so the system is controllable

(b) As we previously mentioned the coefficients of the characteristic polynomial are closely related to the last row of the A^* matrix. Therefore, the CCF of the system is:

$$\vec{z}[t+1] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -8 & -7 \end{bmatrix} \vec{x}[t] + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u[t]$$

Example Solutions Continued

(c) Our system then becomes:

$$\vec{z}[t+1] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ f_1 & f_2 - 8 & f_3 - 7 \end{bmatrix} \vec{x}[t]$$

Which means its characteristic polynomial is :
$\lambda^3 - (f_3 - 7)\lambda^2 - (f_2 - 8)\lambda - f_1 = 0.$
Now, we know the characteristic polynomial should be $\lambda(\lambda - \frac{3}{4})(\lambda + \frac{1}{4})$, so we can equate the two and solve for the feedback vector $\vec{f}^{\,T} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{3}{16} \end{bmatrix}$.

Linearization

Linearization

- Recall that if we have $\frac{dx}{dt} = \lambda x(t) + bu(t)$ we know a solution to this is:

$$x(t) = x(0)e^{\lambda t} + \int_0^t e^{\lambda(t-\tau)} u(\tau)\ d\tau$$

- What if we had $\frac{dx}{dt} = f(x(t)) + bu(t)$, where f is nonlinear (e.g *sin*)?

Linearization

- Recall that if we have $\frac{dx}{dt} = \lambda x(t) + bu(t)$ we know a solution to this is:

$$x(t) = x(0)e^{\lambda t} + \int_0^t e^{\lambda(t-\tau)}u(\tau)\ d\tau$$

- What if we had $\frac{dx}{dt} = f(x(t)) + bu(t)$, where f is nonlinear (e.g *sin*)?
- Big Picture: linearize f around an operating point and then treat it as a linear function in a small neighborhood of that point.
- Why linearization?

Linearization

- Recall that if we have $\frac{dx}{dt} = \lambda x(t) + bu(t)$ we know a solution to this is:

$$x(t) = x(0)e^{\lambda t} + \int_0^t e^{\lambda(t-\tau)} u(\tau)\ d\tau$$

- What if we had $\frac{dx}{dt} = f(x(t)) + bu(t)$, where f is nonlinear (e.g *sin*)?
- Big Picture: linearize f around an operating point and then treat it as a linear function in a small neighborhood of that point.
- Why linearization?
  It allows you to treat the system as a linear one, which is very helpful as linear ODE are (usually) much easier to solve!

Linearizing a Single-Variable Function

- Suppose we have $f(x)$ that is a non linear function. We can use a first order Taylor polynomial to linearize the function, this is equivalent to finding the slope of the tangent line of $f(x)$ at a particular point.
- From calculus: $f(x) \approx f(x^*) + f'(x^*)(x - x^*)$.
- As long as we are within some (very small) δ neighborhood of x^* the linearization is valid.
- Example: Linearize $f(x) = 3e^{x^2+2}$ around x^*

Linearizing a Single-Variable Function

- Suppose we have $f(x)$ that is a non linear function. We can use a first order Taylor polynomial to linearize the function, this is equivalent to finding the slope of the tangent line of $f(x)$ at a particular point.
- From calculus: $f(x) \approx f(x^*) + f'(x^*)(x - x^*)$.
- As long as we are within some (very small) δ neighborhood of x^* the linearization is valid.
- Example: Linearize $f(x) = 3e^{x^2+2}$ around x^*
- Solution:
  $f(x^*) = 3e^{(x^*)^2+2}$
  $f'(x) = 3e^{x^2+2}(2x) = 6xe^{x^2+2}$
  $f'(x^*) = 6x^*e^{(x^*)^2+2}$
  Therefore : $f(x) \approx 3e^{(x^*)^2+2} + 6x^*e^{(x^*)^2+2}(x - x^*)$

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$

(i) Choose, or you may be given, a DC input point. That is, a point $u^* \equiv u(t)$ that is constant with time.

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$

(i) Choose, or you may be given, a DC input point. That is, a point $u^* \equiv u(t)$ that is constant with time.

(ii) Find a DC operating point, $x^* \equiv x(t)$. That is, solve $\frac{dx^*}{dt} = f(x^*) + bu^*$. Notice that this boils down to finding an x^* such that $f(x^*) + bu^* = 0$.

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$

(i) Choose, or you may be given, a DC input point. That is, a point $u^* \equiv u(t)$ that is constant with time.

(ii) Find a DC operating point, $x^* \equiv x(t)$. That is, solve $\frac{dx^*}{dt} = f(x^*) + bu^*$. Notice that this boils down to finding an x^* such that $f(x^*) + bu^* = 0$.

(iii) Define $x_l(t) = x(t) - x^*$ and $u_l(t) = u(t) - u^*$, and re-write the ODE in terms of $x_l(t)$ and $u_l(t)$. By plugging in you get: $\frac{dx_l(t)}{dt} = f(x_l(t) + x^*) + b(u_l(t) + u^*)$

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$

(i) Choose, or you may be given, a DC input point. That is, a point $u^* \equiv u(t)$ that is constant with time.

(ii) Find a DC operating point, $x^* \equiv x(t)$. That is, solve $\frac{dx^*}{dt} = f(x^*) + bu^*$. Notice that this boils down to finding an x^* such that $f(x^*) + bu^* = 0$.

(iii) Define $x_l(t) = x(t) - x^*$ and $u_l(t) = u(t) - u^*$, and re-write the ODE in terms of $x_l(t)$ and $u_l(t)$. By plugging in you get: $\frac{dx_l(t)}{dt} = f(x_l(t) + x^*) + b(u_l(t) + u^*)$

(iv) It is ok to assume at this point that $u_l(t)$ is small, that means that the $u(t)$ in step 1 does not deviate too much from u^*.

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$

(i) Choose, or you may be given, a DC input point. That is, a point $u^* \equiv u(t)$ that is constant with time.

(ii) Find a DC operating point, $x^* \equiv x(t)$. That is, solve $\frac{dx^*}{dt} = f(x^*) + bu^*$. Notice that this boils down to finding an x^* such that $f(x^*) + bu^* = 0$.

(iii) Define $x_l(t) = x(t) - x^*$ and $u_l(t) = u(t) - u^*$, and re-write the ODE in terms of $x_l(t)$ and $u_l(t)$. By plugging in you get: $\frac{dx_l(t)}{dt} = f(x_l(t) + x^*) + b(u_l(t) + u^*)$

(iv) It is ok to assume at this point that $u_l(t)$ is small, that means that the $u(t)$ in step 1 does not deviate too much from u^*.

(v) Linearize the ODE: $f(x_l(t) + x^*) \approx f(x^*) + f'(x^*)x_l(t)$. Here we assume that $x_l(t)$ is also small. This is something that we will need to verify in the next step!

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$ (Continued)

(vi) Plug (vi) back into (iii) and we obtain :
$\frac{dx_l(t)}{dt} \approx f'(x^*)f(x_l(t)) + \cancel{f(x^*)} + bu_l(t) + \cancel{bu^*} =$
$f'(x^*)f(x_l(t)) + bu_l(t)$

(vii) Verify the linearization!
How do we know if the linearization is valid?

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$ (Continued)

(vi) Plug (vi) back into (iii) and we obtain :
$\frac{dx_l(t)}{dt} \approx f'(x^*)f(x_l(t)) + \cancel{f(x^*)} + bu_l(t) + \cancel{bu^*} =$
$f'(x^*)f(x_l(t)) + bu_l(t)$

(vii) Verify the linearization!

How do we know if the linearization is valid? Well, if we have $\frac{dx_l(t)}{dt} = \lambda x_l(t) + bu(t)$ we know the solution doesn't blow up if $\lambda < 0$ as we will have a term $e^{\lambda t}$.

This means that we want $m = f'(x^*) < 0$.

Linearizing Steps for $\frac{dx(t)}{dt} = f(x(t)) + bu(t)$ (Continued)

(vi) Plug (vi) back into (iii) and we obtain :
$\frac{dx_l(t)}{dt} \approx f'(x^*)f(x_l(t)) + \cancel{f(x^*)} + bu_l(t) + \cancel{bu^*} = f'(x^*)f(x_l(t)) + bu_l(t)$

(vii) Verify the linearization!

How do we know if the linearization is valid? Well, if we have $\frac{dx_l(t)}{dt} = \lambda x_l(t) + bu(t)$ we know the solution doesn't blow up if $\lambda < 0$ as we will have a term $e^{\lambda t}$.

This means that we want $m = f'(x^*) < 0$.

So what do we do if $m > 0$?

(vi) Plug (vi) back into (iii) and we obtain :
$\frac{dx_l(t)}{dt} \approx f'(x^*)f(x_l(t)) + \cancel{f(x^*)} + bu_l(t) + \cancel{bu^*} =$
$f'(x^*)f(x_l(t)) + bu_l(t)$

(vii) Verify the linearization!

How do we know if the linearization is valid? Well, if we have $\frac{dx_l(t)}{dt} = \lambda x_l(t) + bu(t)$ we know the solution doesn't blow up if $\lambda < 0$ as we will have a term $e^{\lambda t}$.

This means that we want $m = f'(x^*) < 0$.

So what do we do if $m > 0$?

We need to go back and change our DC operating point x^*

Practice Problem

Linearize $\frac{dx(t)}{dt} = \cos(x(t)) + bu(t)$ about $u^* = 0$.

Practice Problem

Linearize $\frac{dx(t)}{dt} = \cos(x(t)) + bu(t)$ about $u^* = 0$.
Hint: $\cos(x^*) = 0$ has multiple solutions, which means that we can find numerous DC operating points, can you guess which one would result in a stable system ?

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... - 2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... -2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... - 2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

(iv) We assume that $u_l(t)$ is small.

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... - 2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

(iv) We assume that $u_l(t)$ is small.

(v) Linearize the ODE: $\cos(x_l(t) + \frac{\pi}{2}) \approx \cos(\frac{\pi}{2}) - \sin(\frac{\pi}{2})x_l(t)$.

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... - 2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

(iv) We assume that $u_l(t)$ is small.

(v) Linearize the ODE: $\cos(x_l(t) + \frac{\pi}{2}) \approx \cos(\frac{\pi}{2}) - \sin(\frac{\pi}{2})x_l(t)$.

(vi) Plug (v) back into ODE: $\frac{dx_l(t)}{dt} = -x_l(t) + u_l(t)$

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{... - 2, -1, 1, 2, ...\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

(iv) We assume that $u_l(t)$ is small.

(v) Linearize the ODE: $\cos(x_l(t) + \frac{\pi}{2}) \approx \cos(\frac{\pi}{2}) - \sin(\frac{\pi}{2})x_l(t)$.

(vi) Plug (v) back into ODE: $\frac{dx_l(t)}{dt} = -x_l(t) + u_l(t)$

We see that our assumption that $x_l(t)$ is small is indeed satisfied as we will have a e^{-t} term in the solution which means that $x_l(t)$ will decay.

Practice Problem Solution

(i) We were given the DC input, $u^* = 0$

(ii) $\cos(x^*) = 0$, which means that $x^* = k\frac{\pi}{2}$ for $k \in \{\ldots -2, -1, 1, 2, \ldots\}$. We will choose $x^* = \frac{\pi}{2}$

(iii) Let $x_l(t) = x(t) - \frac{\pi}{2}$ and $u_l(t) = u(t) - 0$. By plugging in we get: $\frac{dx_l(t)}{dt} = \cos(x_l(t) + \frac{\pi}{2}) + u_l(t)$

(iv) We assume that $u_l(t)$ is small.

(v) Linearize the ODE: $\cos(x_l(t) + \frac{\pi}{2}) \approx \cos(\frac{\pi}{2}) - \sin(\frac{\pi}{2})x_l(t)$.

(vi) Plug (v) back into ODE: $\frac{dx_l(t)}{dt} = -x_l(t) + u_l(t)$

We see that our assumption that $x_l(t)$ is small is indeed satisfied as we will have a e^{-t} term in the solution which means that $x_l(t)$ will decay.

What if we had chosen a different DC Operating point, say $-\frac{\pi}{2}$?
When we linearize the system we see that the solution will "explode" around that particular DC operating point.

Linearization of Vector Functions

What if we had $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, \vec{u})$? We will see that the process is very similar to the scalar case we just analyzed!

Linearization of Vector Functions

What if we had $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, \vec{u})$? We will see that the process is very similar to the scalar case we just analyzed!
First, let's see how to linearize $\vec{f}(\vec{x})$ around a DC operating point $\vec{x}^*$. Where $\vec{f} \in \mathbb{R}^{n\times 1}$ is a vector of scalar functions.

Linearization of Vector Functions

What if we had $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, \vec{u})$? We will see that the process is very similar to the scalar case we just analyzed!
First, let's see how to linearize $\vec{f}(\vec{x})$ around a DC operating point $\vec{x}^*$. Where $\vec{f} \in \mathbb{R}^{n\times 1}$ is a vector of scalar functions.
The idea is to linearize individually each one of the f_i around the DC operating point.

Linearization of Vector Functions

What if we had $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, \vec{u})$? We will see that the process is very similar to the scalar case we just analyzed!

First, let's see how to linearize $\vec{f}(\vec{x})$ around a DC operating point $\vec{x}^*$. Where $\vec{f} \in \mathbb{R}^{n\times 1}$ is a vector of scalar functions.

The idea is to linearize individually each one of the f_i around the DC operating point.

For example:

$f_1 \approx f_1(\vec{x}^*) + \frac{\partial f_1}{\partial x_1}(\vec{x}^*)(x_1 - x_1^*) + ... + \frac{\partial f_n}{\partial x_1}(\vec{x}^*)(x_n - x_n^*)$

Linearization of Vector Functions

What if we had $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}, \vec{u})$? We will see that the process is very similar to the scalar case we just analyzed!

First, let's see how to linearize $\vec{f}(\vec{x})$ around a DC operating point $\vec{x}^*$. Where $\vec{f} \in \mathbb{R}^{n\times 1}$ is a vector of scalar functions.

The idea is to linearize individually each one of the f_i around the DC operating point.

For example:

$f_1 \approx f_1(\vec{x}^*) + \frac{\partial f_1}{\partial x_1}(\vec{x}^*)(x_1 - x_1^*) + ... + \frac{\partial f_n}{\partial x_1}(\vec{x}^*)(x_n - x_n^*)$

Repeating this for all n functions in $\vec{f}$ we see we get a system of scalar, linearized, multivariate functions which makes you think, wouldn't it be nice to express it in a shorthand matrix notation?

Jacobian Matrix

We can use the Jacobian to express everything nicely and neatly. The Jacobian is the name given to the matrix of partial derivatives of $\vec{f}$, and it is denoted by $J_{\vec{x}}$ or $\nabla_{\vec{x}}\vec{f}$.

Jacobian Matrix

We can use the Jacobian to express everything nicely and neatly.
The Jacobian is the name given to the matrix of partial derivatives of $\vec{f}$, and it is denoted by $J_{\vec{x}}$ or $\nabla_{\vec{x}}\vec{f}$.
Continuing from the previous slide we have:

$$\begin{bmatrix} f_1(\vec{x}) \\ \vdots \\ f_n(\vec{x}) \end{bmatrix} \approx \begin{bmatrix} f_1(\vec{x}^*) \\ \vdots \\ f_n(\vec{x}^*) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}^*) & \dots & \frac{\partial f_1}{\partial x_n}(\vec{x}^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\vec{x}^*) & \dots & \frac{\partial f_n}{\partial x_n}(\vec{x}^*) \end{bmatrix} (\vec{x} - \vec{x}^*)$$

Linearization with Jacobians Example

$$\text{Linearize } \vec{f}(\vec{x}(t)) = \begin{bmatrix} \sin(x_1(t) \times x_2(t)) + 2x_1(t)x_3^2(t) \\ x_3(t)\cos(x_2(t)) + \frac{x_1(t)}{x_3(t)} \\ x_1(t) + 2x_3(t)x_2^3(t) \end{bmatrix} \text{ about } \vec{x}^* = \begin{bmatrix} 0 \\ 2\pi \\ \frac{2\pi}{3} \end{bmatrix}$$

Solutions

Find the Jacobian:

$$\begin{bmatrix} x_2(t)\cos(x_1(t)\times x_2(t)) + 2x_3^2(t) & x_1(t)\cos(x_1(t)\times x_2(t)) & 4x_1(t)x_3(t) \\ \frac{1}{x_3(t)} & -x_3(t)\sin(x_2(t)) & \cos(x_2(t)) - \frac{x_1(t)}{x_3^2(t)} \\ 1 & 6x_3(t)x_2^2(t) & 2x_2^3(t) \end{bmatrix}$$

Evaluate the Jacobian about $\vec{x}^*$:

$$\begin{bmatrix} 5\pi & 0 & 0 \\ \frac{2\pi}{3} & 0 & 1 \\ 1 & 36\pi^3 & 16\pi^3 \end{bmatrix}$$

Linearize:

$$\vec{f}(\vec{x}(t)) \approx \begin{bmatrix} 0 \\ \frac{3\pi}{4} \\ 24\pi^4 \end{bmatrix} + \begin{bmatrix} 5\pi & 0 & 0 \\ \frac{2\pi}{3} & 0 & 1 \\ 1 & 36\pi^3 & 16\pi^3 \end{bmatrix} \begin{bmatrix} x_1(t) - 0 \\ x_2(t) - \frac{3\pi}{4} \\ x_3(t) - 24\pi^4 \end{bmatrix}$$

Steps to Linearize Vector ODE Systems

To linearize $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$ we use a similar procedure as we did for the scalar case.

Steps to Linearize Vector ODE Systems

To linearize $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$ we use a similar procedure as we did for the scalar case.

(i) Solve $\vec{f}(\vec{x}^*, \vec{u}^*) = \vec{0}$ to determine the equilibrium point.

Steps to Linearize Vector ODE Systems

To linearize $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$ we use a similar procedure as we did for the scalar case.

(i) Solve $\vec{f}(\vec{x}^*, \vec{u}^*) = \vec{0}$ to determine the equilibrium point.

(ii) Let $\vec{x}_l(t) = \vec{x}(t) - \vec{x}^*$ and $\vec{u}_l(t) = \vec{u}(t) - \vec{u}^*$

Steps to Linearize Vector ODE Systems

To linearize $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$ we use a similar procedure as we did for the scalar case.

(i) Solve $\vec{f}(\vec{x}^*, \vec{u}^*) = \vec{0}$ to determine the equilibrium point.

(ii) Let $\vec{x}_l(t) = \vec{x}(t) - \vec{x}^*$ and $\vec{u}_l(t) = \vec{u}(t) - \vec{u}^*$

(iii) Linearize $\vec{f}(\vec{x}, \vec{u})$ about $(\vec{x}^*, \vec{u}^*)$. That is:
$\vec{f}(\vec{x}(t), \vec{u}(t)) \approx \vec{f}(\vec{x}^*, \vec{u}^*) + J_{\vec{x}}\vec{x}_l(t) + J_{\vec{u}}\vec{u}_l(t)$

Steps to Linearize Vector ODE Systems

To linearize $\frac{d\vec{x}}{dt} = \vec{f}(\vec{x}(t), \vec{u}(t))$ we use a similar procedure as we did for the scalar case.

(i) Solve $\vec{f}(\vec{x}^*, \vec{u}^*) = \vec{0}$ to determine the equilibrium point.

(ii) Let $\vec{x}_l(t) = \vec{x}(t) - \vec{x}^*$ and $\vec{u}_l(t) = \vec{u}(t) - \vec{u}^*$

(iii) Linearize $\vec{f}(\vec{x}, \vec{u})$ about $(\vec{x}^*, \vec{u}^*)$. That is:
$\vec{f}(\vec{x}(t), \vec{u}(t)) \approx \vec{f}(\vec{x}^*, \vec{u}^*) + J_{\vec{x}}\vec{x}_l(t) + J_{\vec{u}}\vec{u}_l(t)$

(iv) Plug (iv) back into the ODE:
$\frac{d\vec{x}}{dt} \approx \cancel{\vec{f}(\vec{x}^*, \vec{u}^*)} + J_{\vec{x}}\vec{x}_l(t) + J_{\vec{u}}\vec{u}_l(t)$

Linearizing Vector ODE Systems Example

Given a DC input $u^* = 1$, linearize:

$$\frac{d\vec{x}(t)}{dt} = \begin{bmatrix} x_1^2(t) - x_2(t)u(t) \\ x_2^2(t)(1 + x_1(t)) + \sin(\pi x_1(t)u(t)) \end{bmatrix}$$

Solutions

Again, we will do this in steps:

(i) We are given $u^* = 1$

Solutions

Again, we will do this in steps:

(i) We are given $u^* = 1$

(ii) We need to find a DC operating point, this means solving the following system of equations:

$$x_1^{*2} - x_2^* u^* = 0 \tag{1}$$

$$x_2^{*2}(x_1^* + 1) + \sin(\pi x_1^* u^*) = 0 \tag{2}$$

The solution is $x_1^* = -1$ and $x_2^* = 1$.

Solutions

Again, we will do this in steps:

(i) We are given $u^* = 1$

(ii) We need to find a DC operating point, this means solving the following system of equations:

$$x_1^{*2} - x_2^* u^* = 0 \tag{1}$$

$$x_2^{*2}(x_1^* + 1) + \sin(\pi x_1^* u^*) = 0 \tag{2}$$

The solution is $x_1^* = -1$ and $x_2^* = 1$.

(iii) Let $\vec{x}_l(t) = \vec{x}(t) - \vec{x}^*$ and $\vec{u}_l(t) = \vec{u}(t) - \vec{u}^*$

Solutions

Again, we will do this in steps:

(i) We are given $u^* = 1$

(ii) We need to find a DC operating point, this means solving the following system of equations:

$$x_1^{*2} - x_2^* u^* = 0 \tag{1}$$

$$x_2^{*2}(x_1^* + 1) + \sin(\pi x_1^* u^*) = 0 \tag{2}$$

The solution is $x_1^* = -1$ and $x_2^* = 1$.

(iii) Let $\vec{x}_l(t) = \vec{x}(t) - \vec{x}^*$ and $\vec{u}_l(t) = \vec{u}(t) - \vec{u}^*$

(iv) Linearize,

$$\vec{f}(\vec{x}(t), u(t)) \approx \vec{f}(\vec{x}^*, 1) + \begin{bmatrix} -2 & -1 \\ 1 - \pi & 0 \end{bmatrix} \vec{x}_l(t) + \begin{bmatrix} -1 \\ \pi \end{bmatrix} u_l(t)$$

Solutions Continued

(v) Substitute linear approximation back into the system,

$$\frac{d\vec{x}(t)}{dt} \approx \cancel{\vec{f}(\vec{x}^*, 1)} + \begin{bmatrix} -2 & -1 \\ 1-\pi & 0 \end{bmatrix} \vec{x}_l(t) + \begin{bmatrix} -1 \\ \pi \end{bmatrix} u_l(t)$$

Singular Value Decomposition

SVD Theorem

Any matrix $A \in \mathbb{R}^{m \times n}$ can be decomposed into the product of three matrices

$$A = U \Sigma V^T$$
$$U : m \times m$$
$$\Sigma : m \times n$$
$$V^T : n \times n$$

Such that U, V are unitary matrices and Σ only has nonnegative values along its main diagonal.

SVD: Compact Form

We can also express the SVD as

$$
\begin{gathered}
A = \mathcal{U} S \mathcal{V}^T \\
\mathcal{U} : m \times r \\
S : r \times r \\
\mathcal{V}^T : r \times n
\end{gathered}
$$

where r is the rank of A. The compact form matrices maintain properties of the original matrices, but have entries removed whenever they correspond to zero singular values.

SVD: Outer Product Form

Lastly, we can express

$$A = \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^T$$

where $\vec{u}_i, \vec{v}_i$ are the columns of U, V, respectively, and σ_i are corresponding diagonal entry of the matrix Σ

Computing SVD with $A^T A$

$$
\begin{aligned}
A^T A &= U\Sigma V^T V\Sigma^T U^T \\
&= U\Sigma^2 U^T
\end{aligned}
$$

This is an eigen decomposition since Σ^2 is diagonal and $U^{-1} = U^T$. Thus solving for the eigenvalues and eigenvectors of $A^T A$ give $\lambda_i = \sigma_i^2$ with eigenvectors which correspond to the right singular vectors. We need to sort by decreasing σ_i.
Side note: $\Sigma^T \Sigma$ is not actually equal to Σ^2, but the former product yields a matrix with singular values squared on the diagonal entries, hence we call it Σ^2

Computing SVD with $A^T A$

Given a right singular vector $\vec{v}_i$ which we found from the previous part, we can apply it

$$
\begin{aligned}
A\vec{v}_i &= \left(\sum_{k=1}^{r} \sigma_k \vec{u}_k \vec{v}_k^T\right) \vec{v}_i \\
&= \sum_{k=1}^{r} \sigma_k \vec{u}_k \vec{v}_k^T \vec{i} \\
&= \sigma_i \vec{u}_i \\
\vec{u}_i &= \frac{1}{\sigma_i} A\vec{v}_i
\end{aligned}
$$

Computing SVD with AA^T

Similar calculations yield $\sigma_i = \sqrt{\lambda_i}$ of AA^T with eigenvectors as left singular vectors, and $\vec{v}_i = \frac{1}{\sigma_i} A^T \vec{u}_i$

Intepretation of SVD

- Unitary matrices act as rotation in a given space. A diagonal matrix stretches in a given coordinate space.
- SVD visualization (open in browser)

Intepretation of SVD

For a product $A\vec{x}$, we can decompose every vector $\vec{x}$ into a linear combination of right singular vectors

$$\vec{x} = \sum_{i=1}^{n} \alpha_i \vec{v}_i$$

Thus, we can see exactly which parts of $\vec{x}$ affect the output.

Compression of Low-Rank Matrices

- Suppose I had a matrix $A \in \mathbb{R}^{m \times n}$ with $m, n >> rank(A)$. How could I more efficiently store A and compute products like $A\vec{x}$?

Compression of Low-Rank Matrices

- Suppose I had a matrix $A \in \mathbb{R}^{m \times n}$ with $m, n >> rank(A)$. How could I more efficiently store A and compute products like $A\vec{x}$?

- With the SVD, we only have to save r set of two vectors and a scalar, which saves us a lot of space if the rank is small with respect to the matrix. Also, less computation is carried out if we represent the matrix as the outer product form.

Principle Component Analysis

PCA

PCA is a linear dimensionality reduction tool. Given data $\vec{x}_i \in \mathbb{R}^d$, we can create a mapping $T : \mathbb{R}^d \to \mathbb{R}^{d'}, d' < d$ such that the variance in the dataset is still captured

PCA — Computation

1. Store data row-major in $A \in \mathbb{R}^{n \times d}$

PCA — Computation

1. Store data row-major in $A \in \mathbb{R}^{n \times d}$
2. De-mean A

PCA — Computation

1. Store data row-major in $A \in \mathbb{R}^{n \times d}$
2. De-mean A
3. Take SVD: $A = U\Sigma V^T$

PCA — Computation

1. Store data row-major in $A \in \mathbb{R}^{n \times d}$
2. De-mean A
3. Take SVD: $A = U \Sigma V^T$
4. Create $V_{d'} \in \mathbb{R}^{n \times d'}$ from vectors of V corresponding to d' greatest signular values

PCA — Computation

1. Store data row-major in $A \in \mathbb{R}^{n \times d}$
2. De-mean A
3. Take SVD: $A = U \Sigma V^T$
4. Create $V_{d'} \in \mathbb{R}^{n \times d'}$ from vectors of V corresponding to d' greatest signular values
5. To project data into the representative subspace: $T(x) := V_{d'}^T x$

PCA: Computation

The mapping T can then be expressed as

$$T(\vec{x}) = V_k^T \vec{x}$$

If we apply this transformation onto the entire dataset (which has row vectors), we can say

$$T(A) = B = AV_k$$

where $B \in \mathbb{R}^{n \times k}$

PCA: computation

If we were to show the projected vectors in the original space, we can multiply back with the projection vectors

$$A' = BV_k^T$$

Discretization

Discretization: Q1

Note: this section follows hw8 q1 almost exactly. Suppose we have a scalar system

$$\frac{d}{dt}x(t) = \alpha x + \vec{\beta}^T \vec{u}(t)$$

and we apply a constant input $\vec{u}_n$ for times $t \in [nT, (n+1)T)$ for some $T > 0$. Given $x(nT)$ solve the differential equation

Discretization: Q1 Sol

From $t = nT$ to $t = (n+1)T$, $\vec{\beta}^T \vec{u}$ is a constant scalar. Thus, we can solve this like a normal differential equation. Let $x = x' - \frac{\vec{\beta}^T \vec{u}}{\alpha}$.

$$\begin{aligned}
\frac{d}{dt}x(t) &= \alpha(x' - \frac{\vec{\beta}^T \vec{u}}{\alpha}) + \vec{\beta}^T \vec{u}(t) \\
&= \alpha x' \\
x' &= Ae^{\alpha(x-nT)} \\
x + \frac{\vec{\beta}^T \vec{u}}{\alpha} &= Ae^{\alpha(x-nT)} \\
x &= Ae^{\alpha(x-nT)} - \frac{\vec{\beta}^T \vec{u}}{\alpha}
\end{aligned}$$

Discretization: Q1 Sol Continued

At which point we can use our initial condition to get

$$x(nT) = A - \frac{\vec{\beta}^T \vec{u}}{\alpha}$$

$$A = x(nT) + \frac{\vec{\beta}^T \vec{u}}{\alpha}$$

$$x = \left(x(nT) + \frac{\vec{\beta}^T \vec{u}}{\alpha} \right) e^{\alpha(t-nT)} - \frac{\vec{\beta}^T \vec{u}}{\alpha}$$

Discretization: Q2

Using the differential equation derived from question 1, create a discrete-time system to model the continuous time. In other words, if $x[n] = x(nT), \vec{u}[n] = \vec{u}(nT)$, find a relation such that

$$x[n+1] = A_d x[n] + B_d \vec{u}[n]$$

Discretization: Q2 Sol

We can solve the previous solution for $x((n+1)T)$

$$x((n+1)T) = \left(x(nT) + \frac{\vec{\beta}^T \vec{u}(nT)}{\alpha} \right) e^{\alpha((n+1)T - nT)} - \frac{\vec{\beta}^T \vec{u}(nT)}{\alpha}$$

$$x[n+1] = e^{\alpha T} x[n] + \frac{e^{\alpha T} - 1}{\alpha} \vec{\beta}^T \vec{u}[n]$$

We see that $A_d = e^{\alpha T}, B_d = ((e^{\alpha T} - 1)/\alpha)\vec{\beta}^T$

Discretization: Q3

Instead of a scalar, we instead have a diagonal matrix A such that

$$\frac{d}{dt}\vec{x} = A\vec{x} + B\vec{u}$$

Discretize this system in the same was as Q2.

Discretiziation: Q3 Sol

Expanding the original system out line-by-line gives

$$\frac{d}{dt}x_i = a_i x_i + b_i \vec{u}_i$$

where x_i is the ith variable of $\vec{x}$, a_i is the diagonal entry of A, and b_i is the row of B.

Discretization: Generic Matrix

Math not shown, but we can perform a change of basis from our original space to our diagonal space, and then apply the results of the previous part.