Module 05
Discrete Time Systems

Ahmad F. Taha

EE 5143: Linear Systems and Control

Email: ahmad.taha@utsa.edu

Webpage: http://engineering.utsa.edu/ataha

October 3, 2017

From CT to DT Systems

- In the previous module, we discussed the basic idea of discretization
- Basically, how to obtain state-space matrices for discretized representation of CT systems
- This necessitates understanding the calculus of DT systems, starting from the difference equation

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) + Du(k)$$

- Note: A, B, C, D here are assumed to be discretized one (derived from one the discretization methods in the previous module). In other words, they're $\tilde{A}, \tilde{B}, \ldots$
- In many situations, we arrive at a DT system after discretization
- However, in many other situations, DT systems (difference equations) depict the actual physics

DT Systems: An Example

- The problem of compound interest/loan payment is a DT control system
- Suppose you owe \$1000 to a bank at $t = k = 0$, and your monthly interest rate is 1.5%
- Also, suppose that the minimum payment is \$50 and you never pay more than the minimum payment
- Hence, we can write:

$$x(k+1) = 1.015x(k) + u(k), \quad x(0) = 1000$$

- $x(k)$ represents the amount of money you still owe; $u(k) = -50$ is the constant monthly payment
- Question 1: Compute your remaining debt after 10 payments
- Question 2: How long it will take to pay it all off?

Solution

$$x(k+1) = \lambda x(k)+\beta u(k), \quad x(0) = x_0 = 1000, \quad \lambda = 1.015, \beta = 1 \text{ are given}$$

1. $x(1) = \lambda x(0) + \beta u(0), \quad x(2) = \lambda^2 x(0) + \lambda\beta u(0) + \beta u(1)$
   $$x(3) = \lambda^3 x(0) + \lambda^2 \beta u(0) + \lambda\beta u(1) + \beta u(2)$$
2. Hence, one can write: $x(k) = \lambda^k x(0) + \sum_{j=0}^{k-1} \lambda^j \beta u(k-1-j)$
3. For this particular problem, $u(k) = u(j) = -50 = \gamma$, therefore:

$$\boxed{x(k) = \lambda^k x(0) + \beta\gamma \sum_{j=0}^{k-1} \lambda^j = \lambda^k x(0) + \beta\gamma \left(\frac{1-\lambda^k}{1-\lambda}\right)}, \forall k$$

4. Question 1 Solution:
   $$x(10) = (1.015)^{10} \cdot 1000 + (-50 \cdot 1) \left(\frac{1-1.015^{10}}{1-1.015}\right) = \$625.40$$
5. Question 2 Solution: find k such that $x(k) = 0$

$$0 = (1.015)^k 1000 - 50 \left(\frac{1-1.015^k}{1-1.015}\right) \Rightarrow k \approx 23.96 = 24 \text{ payments}$$

DT Systems: Why do we need it?

- As we saw in the previous example, we were able to compute two important quantities via the accurate model of loan payments
- We computed how many monthly payments is needed to pay off the debt
- We can also easily obtain how much is left at any $k < 24$
- This case that we discussed is for the scalar case, i.e.,

$$x(k+1) = \lambda x(k) + \beta u(k)$$

- What if we have n-dimensional state-space, i.e.,

$$x(k+1) = Ax(k) + Bu(k)$$

- How can we find $x(k)$ at any k? How can we find k that would yield $x(k) =$ 0-vector?
- To do that, we need to have theory that supports DT system, in contrast with CT LTI systems and matrix exponentials

State Space of DT LTI Systems

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) + Du(k)$$

- To find $x(k)$, we need to find A^k (analogous to matrix exponentials for the CT case)
- Let's consider that $u(k) = 0$, then it's easy to see that:

$$x(1) = Ax(0), \quad x(2) = Ax(1) = A^2x(0), \quad \Rightarrow \boxed{x(k) = A^k x(0) \Rightarrow y(k) = CA^k x(0)}$$

- How to find A^k? Can you simply raise the entries of A to the k-th power?
- No! You cannot! To find A^k, diagonalize $A = TDT^{-1}$
- Then[1], we can write $A^k = TD^kT^{-1}$
- If the matrix is not diagonalizable, find the Jordan form, $(A^k = TJ^kT^{-1})$
- In that case, $J^k = \begin{bmatrix} \lambda^k & k\lambda^{k-1} \\ 0 & \lambda^k \end{bmatrix}$ for a Jordan block of λ with size 2

[1] We proved that in one of the homeworks.

State Space of DT LTI Systems

- So, what if we have a nonzero control $u(k)$?
- We need to obtain an explicit solution $x(k)$ given $x(0)$ and $u(k)$
- We can prove that for

$$x(k+1) = Ax(k) + Bu(k)$$

the state solution is:

$$\boxed{x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j} Bu(j) = A^k x(0) + \sum_{j=0}^{k-1} A^j Bu(k-1-j) \quad (*)}$$

- This is very similar to $x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$ which we derived before
- Equation $(*)$ can be proved via induction, or even by intuition

Solutions of DT LTI Systems

$$\boxed{x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j} Bu(j) = A^k x(0) + \sum_{j=0}^{k-1} A^j Bu(k-1-j) \quad (*)}$$

- The above equation entails: (a) finding closed form solution to A^k and (b) being clever with summations (instead of integrals)
- Again, as mentioned earlier, to find A^k: either find the diagonal or Jordan canonical forms
- The complexity remains if the summation is difficult to **analytically compute**
- Let's do two examples to demonstrate that
- Notice that there's two ways to compute $x(k)$—look at $(*)$
- This means that you should pick the equation which is easy to analytically evaluate

DT LTI Systems — Example 2

$$\boxed{x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j} Bu(j) = A^k x(0) + \sum_{j=0}^{k-1} A^j Bu(k-1-j) \quad (*)}$$

- Consider this system:

$$x(k+1) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$u(k) = \lambda_1^k, \quad x(0) = 0, \quad \lambda_{1,2} \neq 1, 0, \quad \lambda_1 \neq \lambda_2$$

- Important summation rule 1: $\sum_{j=0}^{k-1} \alpha^j = \dfrac{1-\alpha^k}{1-\alpha}$ assuming that $\alpha \neq 1$
- Find $x(k)$. **Solution:**

$$x(k) = \begin{bmatrix} k\lambda_1^{k-1} \\ \lambda_2^{k-1} \dfrac{1-\left(\frac{\lambda_1}{\lambda_2}\right)^k}{1-\frac{\lambda_1}{\lambda_2}} \end{bmatrix}$$

DT LTI Systems — Example 3

- Consider this system:

$$x(k+1) = \begin{bmatrix} 1 & -0.5 \\ 0.5 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 2 \\ -2 \end{bmatrix} u(k)$$

$$u(k) = 1, \quad x(0) = \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \quad \lambda_{1,2} \neq 1, 0, \quad \lambda_1 \neq \lambda_2$$

- **Important summation rule 2**:

$$\sum_{j=0}^{k-1} j\alpha^{j-1} = \frac{d}{d\alpha} \sum_{j=0}^{k-1} \alpha^j = \frac{d}{d\alpha}\left[\frac{1-\alpha^k}{1-\alpha}\right] = \frac{1 - k\alpha^{k-1} + (k-1)\alpha^k}{(1-\alpha)^2}$$

- Find $x(k)$

Solution to Example 3

$$\boxed{x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j} Bu(j) = A^k x(0) + \sum_{j=0}^{k-1} A^j Bu(k-1-j) \quad (*)}$$

1. First, we can write A as:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} = TJT^{-1}$$

2. Find $A^k = TJ^kT^{-1}$, with $J^k = \begin{bmatrix} 0.5^k & k0.5^{k-1} \\ 0 & 0.5^k \end{bmatrix}$, then:

$$x(k) = TJ^kT^{-1}x(0) + T\sum_{j=0}^{k-1} J^j T^{-1} Bu(k-1-j)$$

3. $T^{-1}Bu(k-1-j) = T^{-1}B = v = \begin{bmatrix} 0 & 2 \end{bmatrix}^\top$ constant, hence:

$$x(k) = TJ^kT^{-1}x(0) + T\left(\sum^{k-1} J^j\right) v$$

Solution to Example 3 — 2

$$x(k) = TJ^kT^{-1}x(0) + T\left(\sum_{j=0}^{k-1} J^j\right)v$$

- The only difficult term to evaluate in the above equation is the summation
- Everything else is given
- Recall that

$$J^k = \begin{bmatrix} 0.5^k & k0.5^{k-1} \\ 0 & 0.5^k \end{bmatrix} \Rightarrow \sum_{j=0}^{k-1} J^j = \sum_{j=0}^{k-1} \begin{bmatrix} 0.5^j & j0.5^{j-1} \\ 0 & 0.5^j \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{k-1} 0.5^j & \sum_{j=0}^{k-1} j0.5^{j-1} \\ 0 & \sum_{j=0}^{k-1} 0.5^j \end{bmatrix}$$

- This matrix has three summations, that can be immediately evaluated via summation rules 1 and 2, then:

$$x(k) = TJ^kT^{-1}x(0) + T\begin{bmatrix} \dfrac{1-0.5^k}{1-0.5} & \dfrac{1-k0.5^{k-1}+(k-1)0.5^k}{(1-0.5)^2} \\ 0 & \dfrac{1-0.5^k}{1-0.5} \end{bmatrix} v$$

Final Solution to Example 3

$$x(k) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0.5^k & k0.5^{k-1} \\ 0 & 0.5^k \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1-0.5^k}{1-0.5} & \dfrac{1-k0.5^{k-1}+(k-1)0.5^k}{(1-0.5)^2} \\ 0 & \frac{1-0.5^k}{1-0.5} \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \left(\begin{bmatrix} 0.5^k & k0.5^{k-1} \\ 0 & 0.5^k \end{bmatrix} + \begin{bmatrix} \frac{1-0.5^k}{1-0.5} & \dfrac{1-k0.5^{k-1}+(k-1)0.5^k}{(1-0.5)^2} \\ 0 & \frac{1-0.5^k}{1-0.5} \end{bmatrix} \right) \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2-0.5^k & 4-3k0.5^{k-1}+4(k-1)0.5^k \\ 0 & 2-0.5^k \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 8-6k0.5^{k-1}+8(k-1)0.5^k \\ 4-2\cdot 0.5^k \end{bmatrix}$$

$$= \begin{bmatrix} 12-6k0.5^{k-1}+(8k-10)0.5^k \\ 4-6k0.5^{k-1}+(8k-6)0.5^k \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Remarks

- What we've done so far is analyze state solutions (and you can easily obtain output solutions) for DT systems
- DT systems emerge naturally from systems where time is discrete
- DT systems also emerge from discretization of CT systems
- If a discretization is computed, and it's very accurate, then $x(k) \approx x(t)$ between two sampling instances

Intro to TV DT Linear Systems

- Previously, we assumed that the system is time invariant

$$x(k+1) = Ax(k) + Bu(k)$$

$$\boxed{x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j} Bu(j) = A^k x(0) + \sum_{j=0}^{k-1} A^j Bu(k-1-j) \quad (*)}$$

- A, B, C, D were all constant matrices for the LTI DT systems
- What if we have the following:

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

- What will the state solution be?
- To do that, let's get some help from the STM for DT systems

STM of DT Systems with no Inputs

STM of DT Autonomous Systems

The STM for DT systems $\boxed{x(k+1) = A(k)x(k)}$ is defined as $\phi(n,k)$ such that for any $x(k)$, we have

$$x(n) = \phi(n,k)x(k)$$

- So what is $\phi(n,k)$ in this case? We can easily derive it:

$x(k+1) = A(k)x(k); \quad x(k+2) = A(k+1)x(k+1) = A(k+1)A(k)x(k), \ldots$

- Hence:

$$\boxed{x(n) = A(n-1)A(n-2)\cdots A(k+1)A(k)x(k) = \left(\prod_{j=k}^{n-1} A(j)\right) x(k)}$$

STM of DT Autonomous Systems—2

For the above system, the STM is $\phi(n,k) = \prod_{j=k}^{n-1} A(j)$

Properties of STM of DT Systems

$$x(k) = \phi(k,0)x(0) + \sum_{j=0}^{k-1} \phi(k, j+1)B(j)u(j)$$

$$\phi(n,k) = \prod_{j=k}^{n-1} A(j) = A(n-1)A(n-2)\cdots A(k+1)A(k)$$

1. For DT LTI systems, $\phi(n,k) = A^{n-k}$
2. For DT LTI systems, if $k = 0$ (i.e., zero ICs), $\phi(n) = A^n$
3. The STM $\phi(n,k)$ can be singular. If $A(k), \forall k$ is nonsingular, then $\phi(n,k)$ is nonsingular
4. $\phi(n,n) = I, \forall n$
5. $\phi(k_3,k_2)\phi(k_2,k_1) = \phi(k_3,k_1), \quad \forall k_3 \geq k_2 \geq k_1$
6. STM satisfy the difference equation:

$$\phi(k+1, j) = A(k)\phi(k, j)$$

TV DT Systems

State Solution of TVDT Systems

The state solution for time-varying DT systems

$$\boxed{x(k+1) = A(k)x(k) + B(k)u(k)}$$

is defined as

$$x(k) = \phi(k,0)x(0) + \sum_{j=0}^{k-1} \phi(k, j+1)B(j)u(j)$$

where $\phi(n,k) = \prod_{j=k}^{n-1} A(j)$.

- Can you prove the above theorem? You can do that by induction
- First, show that the formula is true for $k = 0$. Then, assume it's true for k, and prove it for $k+1$
- You should use the fact that DT systems satisfy the difference equation: $\phi(k+1, j) = A(k)\phi(k, j)$

Example 4

- Consider this dynamical system

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = C(k)x(k) + Du(k)$$

- For this system, only the $C(k)$ matrix is time-varying
- **Question:** Given that you have three sets of input-output data:

$$(y(k), u(k)), (y(k+1), u(k+1)), (y(k+2), u(k+2))$$

  and $x(k)$ is unknown, derive an equation that would allow you to obtain $x(k)$

- **Solution:**

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} = \begin{bmatrix} C(k) \\ C(k+1)A \\ C(k+2)A^2 \end{bmatrix} x(k) + \begin{bmatrix} D & 0 & 0 \\ C(k+1)B & D & 0 \\ C(k+2)AB & C(k+2)B & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix}$$

Example 4 (Cont'd)

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} = \begin{bmatrix} C(k) \\ C(k+1)A \\ C(k+2)A^2 \end{bmatrix} x(k) + \begin{bmatrix} D & 0 & 0 \\ C(k+1)B & D & 0 \\ C(k+2)AB & C(k+2)B & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix}$$

- Given the above equation, and since the input-output data is given, we can write the following:

$$Y = \bar{A}x(k) + \bar{B}U \Rightarrow \boxed{(Y - \bar{B}U) = \bar{A}x(k)}$$

- The LHS of the boxed equation is constant, and the only unknown in this equation is $x(k)$
- How to find $x(k)$?
- This is similar to solving a linear systems of equations: $Ax = b$
- When is this linear system solved for a unique $x(k)$?
- **Answer:** if $\bar{A}$ is full row rank, then $x(k)$ can be obtained

Solution to Rectangular $Ax = b$

$$\boxed{(Y - \bar{B}U) = \bar{A}x(k) \qquad \equiv \qquad b = Ax}$$

- Matrix $\bar{A}$ is a tall-skinny, rectangular matrix
- This equation is similar to solving $Ax = b$ for rectangular $A \in \mathbb{R}^{m \times n}, m > n$
- How to solve this equation? When is there a solution?
- $Ax = b$ has a consistent solution when $\text{rank}[A\ ,\ b] = \text{rank}(A)$
- Or whenever $b \in \text{column-space}(A)$
- The solution is unique if and only if $\text{rank}(A) = n$, i.e., A has full column rank
- The unique solution is given by: $x = A^{-L}b$, where A^{-L} is called the left inverse of A
- A left inverse of A is one that satisfies $A^{-L}A = I$
- Moore-Penrose pseudo left inverse is equal to: $A^{-L} = (A^\top A)^{-1}A^\top$ (Matlab's `pinv` command computes that)
- How did we obtain this?

What if A is not full column rank

$$\boxed{(Y - \bar{B}U) = \bar{A}x(k) \qquad \equiv \qquad b = Ax}$$

- This method can be also generalized for fat matrices with more columns than rows (the left inverse then becomes a right inverse)
- So, after obtaining $x = A^{-L}b = (A^\top A)^{-1}A^\top b$, we get the initial conditions or the needed $x(k)$ given the input-output measurements
- What if the A-matrix ($\bar{A}$) is not full column rank and there's no solution to $Ax = b$?
- Well, we'll have to settle for a **least-squares solution**
- A least squares solution is a one that minimizes the error $b - Ax$
- It solve this problem:

$$\underset{x}{\text{minimize}} \, \|b - Ax\|_2,$$

i.e., find x that minimize the error—a simple optimization problem

Example

- Solve $Ax = b$ for $A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, x \in \mathbb{R}^2$
- Clearly, A is not full column rank
- Solve on Matlab: `x=pinvA*b` $= \begin{bmatrix} 0.75 \\ 0 \end{bmatrix}$—this is a least squares solution, a solution that minimizes the error $b - Ax$
- Now, let's set $A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$—full rank A
- `x=pinvA*b` $= (A^\top A)^{-1} A^\top b = \begin{bmatrix} 0.99 \\ -1.99 \end{bmatrix}$
- `pinv` and the left-inverse yielded the same solution as the equations are consistent and A is full column rank

Linearization of Nonlinear Systems — Unrelated Topic

- We learned previously how to compute equilibrium points for nonlinear system
- Precisely, if $\dot{x}(t) = f(x,u)$, we learned how to obtain the equilibrium solution to this system x_e, u_e such that $f(x_e, u_e) = 0$
- From this equilibrium point, how do we obtain a linearized state space?
- First, recall that The equation for the linearization of a function $f(x,y)$ at a point (x_e, u_e) is:

$$f(x,u) \approx f(x_e,u_e) + \left.\frac{\partial f(x,u)}{\partial x}\right|_{x_e,u_e}(x - x_e) + \left.\frac{\partial f(x,u)}{\partial y}\right|_{x_e,u_e}(u - u_e)$$

- Now, define

$$\tilde{x} = x - x_e, \tilde{u} = u - u_e \quad \left.\frac{\partial f(x,u)}{\partial x}\right|_{x_e,u_e} = A, \quad \left.\frac{\partial f(x,u)}{\partial y}\right|_{x_e,u_e} = B$$

- Then: $\boxed{\dot{\tilde{x}}(t) = f(x,u) - f(x_e,u_e) \approx A\tilde{x}(t) + B\tilde{u}(t)}$

Few Notes on Linearization

- $f(x,u)$ is not scalar–it's a vector of potentially nonlinear functions to be linearized
- (x_e, u_e) are constant and precomputed
- $A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{x_e,u_e}$: is the Jacobian matrix of partial derivatives of the function $f(x,u)$ w.r.t. x. This Jacobian matrix is then evaluated at x_e, u_e (i.e., it's a constant matrix)
- $B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{x_e,u_e}$: is the Jacobian matrix of partial derivatives of the function $f(x,u)$ w.r.t. u, evaluated at (x_e, u_e)
- Similar linearization procedure for output equation $y(t) = h(x,u)$
- Stability of the linearized system depends on the choice of the equilibrium points (there are stable equilibrium points and unstable ones)

Linearization Example 1

- A model for cubic leaf spring is given as follows:

$$m\ddot{z}(t) = -k_1 z(t) - k_2 z^3(t)$$

- **Question 1:** Assume $m = 1$, find the state-space representation of this nonlinear system
  - **Solution:** $\dot{x}(t) = f(x) = \begin{bmatrix} x_2(t) \\ -k_1 x_1(t) - k_2 x_1^3(t) \end{bmatrix}$
- **Question 2:** Find the equilibrium of this nonlinear system
  - **Solution:** Two points: $x_e^{(1)} = [0, 0]$, $x_e^{(2)} = [\pm\sqrt{\frac{-k_1}{k_2}}, 0]$
- **Question 3:** Linearize the system around the equilibrium points
  - **Solution:**

$$\dot{\tilde{x}}(t) = \begin{bmatrix} 0 & 1 \\ -k_1 - 3k_2 x_{e1}^2 & 0 \end{bmatrix} \tilde{x}(t)$$

For $x_e^{(1)}$: $A^{(1)} = \begin{bmatrix} 0 & 1 \\ -k_1 & 0 \end{bmatrix}$. For $x_e^{(2)}$: $A^{(2)} = \begin{bmatrix} 0 & 1 \\ 2k_1 & 0 \end{bmatrix}$

- **Question 4:** Determine the stability of the linearized system for different values of k_1, k_2

Linearization Example 2

- A pendulum model with friction is given as follows:

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = -10sin(x_1(t)) - x_2(t) + u(t)$$

- **Question 1:** Find the state-space representation of this nonlinear system
  - **Solution:**

$$\dot{x}(t) = f(x,u) = \begin{bmatrix} x_2(t) \\ -10sin(x_1(t)) - x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

- **Question 2:** Find the equilibrium of this nonlinear system given that $u_e = 0$
  - **Solution:**
- **Question 3:** Linearize the system around the equilibrium points
  - **Solution:**
- **Question 4:** Determine the stability of the linearized system

Linearization Example 3

- Another Example:

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} \begin{bmatrix} x_2 \\ -2(1+x_1)x_2 - 4x_1^3 + 2u \end{bmatrix}$$

Linearization of DT Nonlinear systems

- We studied linearization of nonlinear CT dynamic systems
- What about the linearization of nonlinear discrete time systems?

$$x(k+1) = f(x,u)$$

- How do we linearize? Exactly the same procedure as before
- Example:

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} ax_1(k) + bx_1(k)x_2(k) + cx_1(k)u_1(k) \\ dx_2(k) \end{bmatrix}$$

- Solution:

Questions And Suggestions?

Thank You!

Please visit

engineering.utsa.edu/ataha

IFF you want to know more ☺