Study Unit 1

LE1 State variable feedback systems

1.1 sums

- E11.3
- E11.5
- P11.1
- P11.14
- P11.16
- AP11.1
- AP11.3

1.2 Controllability

Formal Defenition of controllability: *A system is completely controllable if there exists an unconstrained control u(t) that can transfer any initial state* $x(t_0)$ *To any other desired location* $x(t)$ *in a finite time* $t_0 \leq t \leq T$
Controllability and observability are requirements for a system, so that all the poles of the colesd loop system can be arbitrarily placed in the complex plane. For the system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

we can determine the controllability using the algebraic condition:

$$rank[\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B} \ldots \mathbf{A^{n-1}B}] = n$$

Where $\mathbf{A}$ is a $n \times n$ matrix and $\mathbf{B}$ is an $n \times 1$ matrix for single input systems, or $n \times m$ for multi input systems. For the case of a single input single output system: define

$$\mathbf{P_c} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B} \ldots \mathbf{A^{n-1}B}]$$

Which is an $n \times n$ matrix. If the determinanat of $\mathbf{P_c}$ is nonzero, the system is controllable

1.3 Observability

Formal Defenition of observability:*A system is completely observable if and only if there exists a finite time T such that the intial state $x(0)$ can be determined from the observation history $y(t)$ given the control $u(t), 0 \leq t \leq T$*
Considder the single-input, single-output system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad and \quad y = \mathbf{C}\mathbf{x}$$

Where $\mathbf{C}$ is a $1 \times n$ row vector and $\mathbf{x}$ is a $n \times 1$ column vector. The system is completely observable when the determinant of the **observability matrix**$\mathbf{P_O}$ is nonzero when:

$$\mathbf{P_O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA^{n-1}} \end{bmatrix}$$

Which is an $n \times n$ matrix

1.4 Pole placement by means of state feedback

Remember from algerbra 2 that if a set of differential equations in the form $\mathbf{x}' = A\mathbf{x}$ then the time response of the system is given by $\mathbf{x}(t) = \mathbf{v}e^{\lambda t}$

1.5 Ackerman'n formula

Way to calculate K with less variables. Set $u = -k\mathbf{x}$ and let $q(\lambda)$ be the desired charachteristic eqn. (as dictated by P.O ans Ts) then:

$$\mathbf{k} = [01 \ldots 1]\mathbf{P}^{-}1q(A);$$

Type A into calculator and apply q(A). Multiply [0 1] first since this is easier

Study Unit 2

LE2 Mathematical models of simple linear systems

2.1 sums

- 1-1
- 1-2
- 1-3
- 1-4
- 1-6
- 1-7
- 1-12

Study Unit 3

LE3 The Z transform

3.1 sums

- 2-1
- 2-2
- 2-3
- 2-4
- 2-6
- 2-7
- 2-9(a,c)
- 2-11
- 2-12
- 2-14
- 2-16
- 2-17
- 2-18
- 2-19
- 2-23
- 2-24
- 2-25
- 2-28
- 2-31
- 2-32

3

3.2 Refresher on Mason's rule

`https://en.wikipedia.org/wiki/Mason's_gain_formula}` Type this out if you have some time later...

3.2.1 Discrete Time

3.2.2 refresher on partial fractions

$$T(z) = \frac{z}{(z+a)(z+b)(z+c)}$$
$$T(z) = \frac{A}{z+a} + \frac{B}{z+b} + \frac{C}{z+c}$$
$$A = \left.\frac{z}{(z+b)(z+c)}\right|_{z=-a}$$
$$B = \left.\frac{z}{(z+a)(z+c)}\right|_{z=-b}$$
$$C = \left.\frac{z}{(z+a)(z+b)}\right|_{z=-c}$$

3.3 Properties of the Z transform

3.3.1 Additition and subtraction

$$\mathfrak{z}[e_1(k) \pm e_2(k)] = E_1(z) \pm E_2(Z)$$

3.3.2 Multiplication by a constant

$$\mathfrak{z}[ae(k)] = a\zeta[e(k)] = aE(z)$$

3.3.3 Real translation

$$\mathfrak{z}[e(k-n)u(k-n)] = z^{-n}E(z)$$

3.3.4 Complex translation

$$\mathfrak{z}[\epsilon^{ak}e(k)] = E(z\epsilon^{-a})$$

3.3.5 Initial Value

$$e(0) = \lim_{z\to\infty} E(z)$$

3.3.6 Final Value

$$\lim_{n\to\infty} = \lim_{z\to 1}(z-1)E(z)$$

3.4 Difference equations

NOTE!!!! study the following methods, power series was asked in ST1.

- Classical approach
- sequential procedure
- Z-transform

  Power series method

  Partial Fraction Method

  Inversion formula Method

  Discrete convolution

- The unit step function transforms to $\frac{z}{z-1}$
- The unit step function is delayed in one example in the textbook and transforms to $\frac{z}{z-1}$:

$$Z\{\delta(k-1)\} = z^{-1}\frac{z}{z-1}$$
$$Z\{\delta(k-1)\} = \frac{1}{z-1}$$

- most other transforms are in the form $c\frac{z}{z-a}$ and they transform to ca^k
- The transform of a shifted series is in the form:

$$Z\{e(k+n)u(k)\} = z^n\left[E(Z) - \sum_{k=0}^{n-1} e(k)z^{-k}\right]$$
$$e.g.$$
$$Z\{e(k-2)u(k)\} = z^{-2}E(z) - ze(1) - ze(0)$$

- sometimes the initial conditions are made zero, which makes:

$$Z\{e(k+n)u(k)\} = z^nE(z)$$
$$also:$$
$$Z\{e(k+n)u(k+n)\} = z^nE(z)$$

Regardless of initial conditions

- In some cases zero initial conditions are assuumed, but I may be confused
- So:

  Compute Z transforms

  Factorize and isolate the appropriate function (usually X(z) or Y(z))

  Apply partial fractions

  Compute inverse Z transforms to get a function of k

3.5 Simulation diagram, signal flow diagrams & State models

Basically the same as in control 1. The symbol 'T' means a delay: $x(k) \rightarrow [T] \rightarrow x(k-1)$ Mason's rule can be applied and state space models are derived in the same way.

3.6 Transfer Functions

Trasfer functions are a function of Z, and can be derived from the difference equation using the Z transform. In signal flow diagrams, [T] besomes $\frac{1}{z}$

Study Unit 4

LE4 Sampling and reconstruction

4.1 sums

- 3-3
- 3-4(a,b)
- 3-5
- 3-8
- 3-9
- 3-12
- 3-13
- 3-20
- 3-22

4.2 difintion of the star transform

section 3.3, examples 3.1 and 3.2 are NB

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs}$$

4.3 Properties of $E^*(s)$

4.3.1 Property 1: E*(s) is periodic in s with period $j\omega$

$$E^*(s + jm\omega_s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \qquad = E^*(s)$$

4.3.2 Property 2: if $E(S)$ has a pole at $s = s_1$, then $E^*(s)$ must have poles at $s = s_1 + jm\omega_s, m = 0, \pm 1, \pm 2 \ldots$

4.3.3 Shannon's sampling theorem

A function of time $e(t)$ which contains no frequency components greater than f_o hertz is uniquely determined by the values of $e(t)$ at any set of sampling points spaced $1/(2f_0)$ seconds apart

Study Unit 5

LE5 Pulse transfer functions for open loop systems

5.1 sums

- 4-1
- 4-2
- 4-3
- 4-4
- 4-6
- 4-9
- 4-11
- 4-13
- 4-14
- 4-15
- 4-16
- 4-18
- 4-24

5.2 z-transfrom methods(study each)

- power series method
- partial fraction method
- inversion formula
- discrete convolution

5.3 The relationship between E(z) and E*(s)

example 4.1, 4.2 and 4.3 are important, you should be able to calc the DC gain of a system

$$\begin{aligned} \mathfrak{z}[e_k)] = E(z) &= e(0) + e(1)z^{-1} + e(2)z^{-2} \dots \\ E^*(s) &= e(0) + e(T)\epsilon^{-Ts} + e(2T)\epsilon^{-2Ts} \dots \\ E(z) &= E^*(s)|_{\epsilon^{sT}=z} \end{aligned}$$

5.4 The pulse transfer function

example 4.4 is important

5.5 Open loop system with a digital filter

5.6 The modified Z transform and time delays

Study Unit 6

LE6 Pulse transfer funcions for closed loop systems

6.1 sums

- 5-1
- 5-2
- 5-3
- 5-4
- 5-6
- 5-9
- 5-11
- 5-14
- 5-16

Study Unit 7

LE7 Time response charachteristics of discrete time systems

7.1 sums

- 6-1
- 6-2
- 6-4
- 6-6
- 6-10
- 6-12
- 6-15
- 6-19

7.2 Discrete system time response

7.3 System charachteristics

7.4 equation

7.5 Mapping of s to z plane

Study Unit 8

LE8 Stability of digital systems

8.1 sums

- 7-2
- 7-3
- 7-4
- 7-6
- 7-7
- 7-8
- 7-12
- 7-20
- 7-21
- 7-22item

8.2 Stability

8.3 Bi-linear transfromation end routh herwitz

8.4 Jury stability

8.5 The root locus

8.6 The bode diagram

Study Unit 9

LE9 Digital controllers

9.1 sums

- 8-1
- 8-2
- 8-8
- 8-9
- 8-24
- 8-25
- 8-26
- 8-27
- 8-28

9.2 Phase lag compensation

9.3 Phase lead compensation

9.4 Lead-lag compensastion

9.5 PID control

Study Unit 10

LE10 Neural networks and fuzzy logic

10.1 Artificial Neural Networks

10.1.1 Origin

- Based on the concept of a human brain
- Uses many, interconnected neurons
- Developed in an attempt to replicate biological learning]i

10.1.2 Functionality

- Networks Useful for:

  Multivariable;

  Time varying;

  Nonlinear systems
- Can predict outcomes using a test set.

10.1.3 structure

Usually consists of an input layer, hidden layers and an output layer. Weights are indexed by layer and neuron. This multidimensional vector of weights is what is optimized. Complex behavior is best modeled using a large amount of simple neurons, while simple behavior is best moddeled using a large number of simple ones.

10.1.4 Training

Training is accomplished by optimizing the weight vector so that a series of inputs result in an output with a minimum error compared to the data associated with the input. This is known as the training set, and care must be taken to avoid bias caused by training the system with a spesific dataset in mind.

Training a single neuron Is an example of linear regression and can be done analytically, the system as a whole, however is not linear

Gradient Descent Defining the error as a function, change the weights to move in the direction of biggest downward gradient until the gradient is zero. Randomization can be advantageous, but the algorithm could return a local minimum. The gradient is computed by taking the numeric partial derivative for a small change in a given weight.

random weights Reliable, avoids user bias and getting stuck in local minima. Does not find the global minimum on its own.

Genetic algorithms Defines the effect of a weight change as a 'mutation', and can use this info to iterate weights more predictably

Back propagation algorithm The error is computed and fed into the system.

10.2 Fuzzy Logic

This branch of soft computing enables digital operations on values that have degrees of uncertainty. These variables are best described linguistically, for example 'warm' or 'tall'. These computations are achieved using Fuzzy sets, and Fuzzy rule bases. Fuzzy sets are logical sets (Think Venn diagram) that allow a value to belong to a set in degrees. This definition of the data is then used to apply a set of fuzzy rules, best described as if-then statements. Some variations of fuzzy logic accept or output numerical data, but they all combine numeric data(the rule base) and linguistic data(the fuzzy set)

10.2.1 Origin

Fuzzy logic was first popularized in the east, and to this day most of the advances in the field are made in japan, China and India. In eastern culture, the concept of uncertainty does not have as negative a connotation and the system was developed to quantify degrees of uncertainty. As with other forms of soft computing, fuzzy logic can be optimized using data sets, which if often easier than optimizing a model from first principles.

10.2.2 Structure

Three structures are mentioned in the notes:

Pure fuzzy logic system Both inputs and outputs are fuzzy sets

Takagi & Sugeno Input is a fuzzy set, output is a numerical function

Fuzzifier/Defuzzifier numerical input and output, is converted before and after going through a pure fuzzy system.

10.2.3 Functionality

Used extensively in control systems. The notes refer to:

- Videography
- Air conditioning
- Washing machines
- Train schedules

The advantages of soft computing hold, so it can be used for nonlinear, multi-variable, time variant systems.

10.2.4 Training

A fuzzy dataset is chosen as the training set. A rule base is defined, and then refined to match the dataset. The system should then predict the outcomes of similar data with a degree of accuracy.