NAME: SHOBHIT GUMBER
SID: 12143424

Final Exam

SID: 12143424

Q1) C = 4×10 = 40

D = 100 − C − 20
= 100 − 40 − 20 = 40

E = 2×10 = 20

F = 100 − E − 10
= 100 − 20 − 10
= 70

C = 40
D = 40
E = 20
F = 70

(a) Linear discrete time system.

$x_1(0) = 20{,}000$ $x_2(0) = 8{,}000$ $x_3(0) = 22{,}000$

$u(k) = [u_1(k) + u_2(k)]^T$ $x(k) = [x(k), x_2(k), x_3(k)]^T$

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ x(0) = x_0 \end{cases}$$ for linear discrete systems

where $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in R^n$

$x_1 = 20{,}000$ $x_2 = 8{,}000$ $x_3 = 22{,}000$.

(b) C = 40% D = 40% E = 20% F = 70%

C_1 = 40% × 20,000
= 8000

D = 40% × 20000
= 8000

Assuming all values are correct

$E_x(k)$ = 20% × 8000
= 1600

F(k) = 70% × 8000
= 5600

$$x(k+1) = \begin{bmatrix} 8000 & 8000 \\ 1600 & 5600 \end{bmatrix} x(k) + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} u(k)$$

1(b) Solution - General

$$x(k) = A^k x_0 + \sum_{i=0}^{k-1} A^i Bu(k-1-i)$$

insted of y_i used actual values

$$= \begin{bmatrix} 8000 & 8000 \\ 1600 & 5600 \end{bmatrix}^k x_0 + \sum_{i=0}^{k-1} \begin{bmatrix} 8000 & 8000 \\ 1600 & 5600 \end{bmatrix}^i \times \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}_u \times (k-1-i)$$

Nat Soln

(c) Stability of a linearly discrete system is similar to that of a continuos system however, is only stable if the eigen value is less than 1.

- Using A matrix
- Since the natural response of $x(k+1) = Ax(k) + Bu(k)$ is $x(k) = A^k x_0$

if the initial $|\alpha_i| > 1$, the system is unstable but is stable if $|\alpha_i| \leq 1$

Assuming 1(b) is correct
A is analysed
Using MATLAB and confirming using det
$\det |\lambda I - A| = 0$.

$$(\lambda - 8000) \times (\lambda - 5600) - (8000 \times 1600)$$

$$\text{so } \lambda = 1.0574 \text{ or } 0.3026.$$

Since the poles on the system are at $\underline{1.0574} > 1$ & over. and $0.3026 < 1$

However since 1.0574 is extremely close to 1 it could be said that the system is managable but still unstable. However pole placement can be used to make it stable. Furthermore, our tutor suggested if the eigenvalue is 0, the system is not stable. Therefore can be argued

(d) ~~As stated by our tutor if~~

(a) If the answer in 1(c) is correct and the system is unstable a row rank needs to be determined using the formula.

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

However since b has unknown values ~~A neat trick discussed by our tutor included that if the eigen values have a value of 0 the system is uncontrollable. Therefore it can be said that this linearly discrete system is controllable~~

if computed it would have 2 unknown values. However it is known that if the determinate = 0 then the system is not full rank.

using matlab the rank was found -8000 or $C \neq 0$ ∴ system is controllable. Was too late to change matrix + answers

(e) Linear Quadratic regulator uses algebric Riccati equation:-

$$P_{ss} = Q + A^T P_{ss} A - A^T P_{ss} B (R + B^T P_{ss} B)^{-1} B^T P_{ss} A$$

- P can be found by itterating the ricatti recursion, or by direct methods.

- LQR optimal input is approx a linear constant state feedback

* $u_t = K_{ss} x_t$, $K_{ss} = -(R + B^T P_{ss} B)^{-1} B^T P_{ss} A$.

- It is widely used in practice.

* In simple steps using LQR can be done easily by finding P which would allow to find K. K can then be used to find u_t.

Q2)

ⓐ Image <u>Resolution:-</u> 1280x1024

Principle point = (640,512)

focal length = (950,950)

$R = I$ (No rotation)

$T = [5, 15, 3]$

$Z = 1.5m.$

SID: 12143424.

$\underbrace{121}_{A}43\underbrace{424}_{B}$

$A = 121 + 400 = 521$ $B = 424 + 500 = 924.$

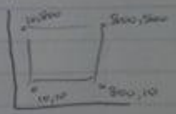
current

~~desired~~ location = [25,80] [521,25] [80,924] [521,924]

desired location = (10,10) (800,10) (10,800) (800,800)

$\lambda = 0.15$ - error correction.

- first we see if the co-ordinate is within the camera's cordinate frame.



- we Minimise error using λ.

$e(t) = s(m(t), a) - s^*$ $\qquad e(t) = -\lambda e.$

where $m(t)$ is feature measurements.

$s^* \rightarrow$ desired values

$s(m(t), a) \rightarrow$ calculated value

from week 9 MATLAB script example.

$$Obs_{xy} = \frac{Obs - [P_x \ P_y]}{f}$$

$$Target\ xy = \frac{Target - [P_x \ P_y]}{f}$$

Also known $\dot{s} = L_s V_c.$ ↓ feature Jacobian.

V_c comp = (V_c, ω_c)

we know $\dot{e} = \underbrace{L_e}_{= L_s} \cdot V_e$

Then least squares are used

$V_c = -\lambda L_e^+ e$

where

$L_e^+ = (L_e^T L_e)^{-1} L_e^T$ ← pseudo-inverse of L_e

Q2a cont.

Since four features are used. we use L_x functions against features.

$$L_x = \begin{bmatrix} -1/z & 0 & x/z & xy & -(1+x^2) & y \\ 0 & -1/z & y/z & 1+y^2 & -xy & -x \end{bmatrix}$$

- After changing values and computing velocities these results are produced.

used $z = 1500$ mm.

$$v_c = -\lambda L_e^+ e \quad = \quad v_c = \begin{bmatrix} -132.4129 \text{ mm/s} \\ 71.7080 \text{ mm/s} \\ 11.1398 \text{ mm/s} \\ 0.0309 \\ 0.0555 \\ 0.0017 \end{bmatrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \omega \text{ for rotation matrix.}$$

Q2 (a) Hand eye calibration requires accurate estimation ~~of the~~ between the robot hand/end effector and the optical frame of the camera affixed to the end effector. The problem can be formulated as $AX = XB$ where A and B are robotic arm and camera poses between time frames. X is the unknown transform between the robot hand and transform.

(b) The estimation of a homogenous tranformation from a robot base and world cordinate frame can be obtained by as a byproduct solution by using Robotic Hand Eye calibration formulated by $Ax = ZB$. In this X is defined as the transformation from robot to world cordinate and Z can be derived from the center point to camera frame.