Connor W. Colombo
ID: cwcolomb

Q1.)

let: $\begin{cases} \dot{x} = AX + Bv \\ y = Cv \end{cases}$ | $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$

let: $P = \left[B \,|\, AB \,|\, A^2B \right]$, $Q = \begin{bmatrix} C \\ \hline CA \\ \hline CA^2 \end{bmatrix}$

$\Rightarrow P = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 3 \end{array}\right]$, $Q = \begin{bmatrix} 1 & 2 & 1 \\ \hline -1 & -2 & -1 \\ \hline 1 & 2 & 1 \end{bmatrix}$

$\because$ P has 3 lin. indep. rows,
rank(P) = 3 = n $\longrightarrow$ System is controllable

$\because$ all cols in Q are multiples of each other,
rank(Q) = 1 < n $\longrightarrow$ System is not observable

Q2.) let: $\begin{cases} \dot{X} = AX + Bv \\ Y = CX \end{cases}$ | $A = \begin{bmatrix} 2 & 1 & & & & & \\ 0 & 2 & & & & & \\ & & 2 & & & & \\ & & & 2 & & & \\ & & & & 1 & 1 & \\ & & & & 0 & 1 & \\ & & & & & & 1 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$, $C = \begin{bmatrix} 2 & 2 & 1 & 3 & -1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

let: $J = M^{-1}AM$, $\hat{B} = M^{-1}B$, $\hat{C} = CM$

$\because$ A is in Jordan Form, $M = I = M^{-1}$

- System is controllable iff $\hat{B}^{\lambda_i}$ is full rank $\forall \lambda_i$ | $\hat{B}^{\lambda_i}$ is matrix of last rows of each Jordan Block for λ_i
- System is controllable iff $\hat{C}^{\lambda_i}$ is full rank $\forall \lambda_i$ | $\hat{C}^{\lambda_i}$ is matrix of first columns of each Jordan Block for λ_i

$\because J = \begin{bmatrix} 2 & 1 & & & & & \\ 0 & 2 & & & & & \\ & & 2 & & & & \\ & & & 2 & & & \\ & & & & 1 & 1 & \\ & & & & 0 & 1 & \\ & & & & & & 1 \end{bmatrix}$, $\hat{B} = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

last row

— Divides Eigenvalues

— Divides Jordan Blocks

$\hat{C} = \begin{bmatrix} 2 & 2 & 1 & 3 & -1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$

$\therefore$ for $\lambda_1 = 2$, $\hat{B}^{\lambda_1} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \end{bmatrix}$, $\because \text{rank}(\hat{B}^{\lambda_1}) = 3 = n_1$, λ_1 is controllable | $\hat{C}^{\lambda_1} = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}$, is full rank $\therefore \lambda_1$ is observable

$\therefore$ for $\lambda_2 = 1$, $\hat{B}^{\lambda_2} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$, $\because \text{rank}(\hat{B}^{\lambda_2}) = 2 = n_1$, λ_2 is controllable | $\hat{C}^{\lambda_2} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$, is full rank $\therefore \lambda_2$ is observable

$\because$ all λ_i are controllable, the system is controllable | $\therefore$ the system is observable, b/c all λ_i are

Q3.) FROM LAST WEEK:

$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\dot{X} = AX + Bu$ | $A = \begin{bmatrix} -\alpha & 0 \\ \alpha & -\beta \end{bmatrix}$ $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $A \in \mathbb{R}^{n \times n} \Rightarrow n = 2$

let: $P = [B \,|\, AB \,|\, A^2B]$, $P = \left[\begin{array}{c|c|c} 1 & -\alpha & \alpha^2 \\ 0 & \alpha & -\alpha^2 - \alpha\beta \end{array}\right]$

b/c of the first column, P is full rank $(= n)$ for all α, β $\Rightarrow$ $\boxed{\text{the system is controllable}}$

---

let: $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\dot{X} = AX + Bu$ | $A = \begin{bmatrix} -\alpha & 0 \\ \alpha & -\beta \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

- Intuitively, the system shouldn't be controllable b/c there's no way for the water from the spout (the input) to directly or indirectly get into the upper tank (it can't affect one of the states)

This can be proven by:

let: $P = [B \,|\, AB \,|\, A^2B]$, $P = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ 1 & -\beta & \beta^2 \end{array}\right]$, $\because$ the first row is $\emptyset$, rank(P) = 1 < n

$\therefore$ $\boxed{\text{the system is not controllable}}$

Q4.)

1.a.)

let: $Av=b$

$$A=\begin{bmatrix}1&1&1\\1&2&3\\1&3&2\end{bmatrix},\ v=\begin{bmatrix}x\\y\\z\end{bmatrix},\ b=\begin{bmatrix}3\\0\\3\end{bmatrix}$$

let: $G=[A|b]$

$$\Rightarrow G=\left[\begin{array}{ccc|c}1&1&1&3\\1&2&3&0\\1&3&2&3\end{array}\right]\xrightarrow{\text{swap } R_1,R_2}\left[\begin{array}{ccc|c}1&1&1&3\\1&3&2&3\\1&2&3&0\end{array}\right]$$

$$\xrightarrow{R_2=R_2-R_1}\left[\begin{array}{ccc|c}1&1&1&3\\0&2&1&0\\1&2&3&0\end{array}\right]$$

$$\xrightarrow{R_3=R_3-R_1}\left[\begin{array}{ccc|c}1&1&1&3\\0&2&1&0\\0&1&2&-3\end{array}\right]$$

$$\xrightarrow{R_3=R_3-\frac{1}{2}R_2}\left[\begin{array}{ccc|c}1&1&1&3\\0&2&1&0\\0&0&3/2&-3\end{array}\right]$$

$$\xrightarrow[R_3=\frac{2}{3}R_3]{R_2=\frac{1}{2}R_2}\left[\begin{array}{ccc|c}1&1&1&3\\0&1&1/2&0\\0&0&1&-2\end{array}\right]$$

$$\xrightarrow[R_2=R_2-\frac{1}{2}R_1]{R_1=R_1-R_2}\left[\begin{array}{ccc|c}1&0&1/2&3\\0&1&0&1\\0&0&1&-2\end{array}\right]$$

$$\xrightarrow{R_1=R_1-\frac{1}{2}R_3}\left[\begin{array}{ccc|c}1&0&0&4\\0&1&0&1\\0&0&1&-2\end{array}\right]$$

$$\therefore \boxed{\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}4\\1\\-2\end{bmatrix}}$$

1.b.)

$$A=\begin{bmatrix}1&2&-1\\2&5&-1\\1&3&2\end{bmatrix},\ v=\begin{bmatrix}x\\y\\z\end{bmatrix},\ b=\begin{bmatrix}1\\3\\6\end{bmatrix}$$

let: $G=[A|b]$

$$\Rightarrow G=\left[\begin{array}{ccc|c}1&2&-1&1\\2&5&-1&3\\1&3&2&6\end{array}\right]\xrightarrow{\text{swap } R_1,R_2}\left[\begin{array}{ccc|c}2&5&-1&3\\1&2&-1&1\\1&3&2&6\end{array}\right]$$

$$\xrightarrow{R_2=R_2-\frac{1}{2}R_1}\left[\begin{array}{ccc|c}2&5&-1&3\\0&\frac{-1}{2}&\frac{-1}{2}&\frac{-1}{2}\\1&3&2&6\end{array}\right]$$

$$\xrightarrow{R_3=R_3-\frac{1}{2}R_1}\left[\begin{array}{ccc|c}2&5&-1&3\\0&\frac{-1}{2}&\frac{-1}{2}&\frac{-1}{2}\\0&1/2&5/2&9/2\end{array}\right]$$

$$\xrightarrow{R_3=R_3-R_2}\left[\begin{array}{ccc|c}2&5&-1&3\\0&\frac{-1}{2}&\frac{-1}{2}&\frac{-1}{2}\\0&0&2&4\end{array}\right]$$

$$\xrightarrow[R_2=-2R_2,\ R_3=\frac{1}{2}R_3]{R_1=\frac{1}{2}R_1,}\left[\begin{array}{ccc|c}1&5/2&\frac{-1}{2}&3/2\\0&1&1&1\\0&0&1&2\end{array}\right]$$

$$\xrightarrow[R_2=R_2-R_3]{R_1=R_1-\frac{5}{2}R_2}\left[\begin{array}{ccc|c}1&0&-3&-1\\0&1&0&-1\\0&0&1&2\end{array}\right]$$

$$\xrightarrow{R_1=R_1+3R_3}\left[\begin{array}{ccc|c}1&0&0&5\\0&1&0&-1\\0&0&1&2\end{array}\right]$$

$$\therefore \boxed{\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}5\\-1\\2\end{bmatrix}}$$

1.c.)

$$A=\begin{bmatrix}1&1&-1&1\\2&3&1&0\\3&5&3&-1\end{bmatrix},\ v=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix},\ b=\begin{bmatrix}1\\4\\5\end{bmatrix}$$

let: $G=[A|b]$

$$\Rightarrow G=\left[\begin{array}{cccc|c}1&1&-1&1&1\\2&3&1&0&4\\3&5&3&-1&5\end{array}\right]\xrightarrow{\text{swap } R_3,R_1}\left[\begin{array}{cccc|c}3&5&3&-1&5\\2&3&1&0&4\\1&1&-1&1&1\end{array}\right]$$

$$\xrightarrow{R_1=\frac{1}{3}R_1}\left[\begin{array}{cccc|c}1&\frac{5}{3}&1&\frac{-1}{3}&\frac{5}{3}\\2&3&1&0&4\\1&1&-1&1&1\end{array}\right]$$

$$\xrightarrow[R_3=R_3-R_1]{R_2=R_2-2R_1}\left[\begin{array}{cccc|c}1&\frac{5}{3}&1&\frac{-1}{3}&\frac{5}{3}\\0&-1/3&-1&2/3&2/3\\0&-2/3&-2&4/3&-2/3\end{array}\right]$$

$$\xrightarrow{R_3=R_3-2R_2}\left[\begin{array}{cccc|c}1&\frac{5}{3}&1&\frac{-1}{3}&\frac{5}{3}\\0&-1/3&-1&2/3&2/3\\0&0&0&0&-2\end{array}\right]$$

$\because 0\neq-2$, there doesn't exist a solution for this underdetermined system

4.2.) let: $Ax=b$,

$$A=\begin{bmatrix}1&2&4\\3&8&14\\2&6&13\end{bmatrix},\ x=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix},\ b=\begin{bmatrix}3\\13\\4\end{bmatrix}$$

let: $A=LU$

let: $L_0=I_3,\ U_0=A \Rightarrow L_0=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\ U_0=\begin{bmatrix}1&2&4\\3&8&14\\2&6&13\end{bmatrix}$

$$\xrightarrow[U_2=U_2-L_{21}U_1]{L_{21}=\frac{3}{1}}\begin{bmatrix}1&0&0\\3&1&0\\0&0&1\end{bmatrix},\ \begin{bmatrix}1&2&4\\0&2&2\\2&6&13\end{bmatrix}$$

$$\xrightarrow[U_3=U_3-L_{31}U_1]{L_{31}=\frac{2}{1}}\begin{bmatrix}1&0&0\\3&1&0\\2&0&1\end{bmatrix},\ \begin{bmatrix}1&2&4\\0&2&2\\0&2&5\end{bmatrix}$$

$$\xrightarrow[U_3=U_3-L_{32}U_2]{L_{32}=\frac{2}{2}}\begin{bmatrix}1&0&0\\3&1&0\\2&1&1\end{bmatrix},\ \begin{bmatrix}1&2&4\\0&2&2\\0&0&3\end{bmatrix}$$

$$\therefore L=\begin{bmatrix}1&0&0\\3&1&0\\2&1&1\end{bmatrix},\ U=\begin{bmatrix}1&2&4\\0&2&2\\0&0&3\end{bmatrix}$$

$$\because L=\begin{bmatrix}1&0&0\\3&1&0\\2&1&1\end{bmatrix},\ U=\begin{bmatrix}1&2&4\\0&2&2\\0&0&3\end{bmatrix}$$

let: $Ld=b$

$$\Rightarrow\begin{bmatrix}1&0&0\\3&1&0\\2&1&1\end{bmatrix}\begin{bmatrix}d_1\\d_2\\d_3\end{bmatrix}=\begin{bmatrix}3\\13\\4\end{bmatrix}$$

$$\Rightarrow\begin{cases}d_1=3,\\d_2=13-3d_1=4\\d_3=4-2d_1-d_2=-6\end{cases}$$

$$\Rightarrow d=\begin{bmatrix}3\\4\\-6\end{bmatrix}$$

let: $Ux=d$

$$\begin{bmatrix}1&2&4\\0&2&2\\0&0&3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}3\\4\\-6\end{bmatrix}\Rightarrow\begin{cases}x_3=\frac{1}{3}\cdot-6=-2,\\x_2=\frac{1}{2}(4-2x_3)=4,\\x_1=3-2x_2-4x_3=3\end{cases}$$

$$\therefore \boxed{\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}3\\4\\-2\end{bmatrix}}$$

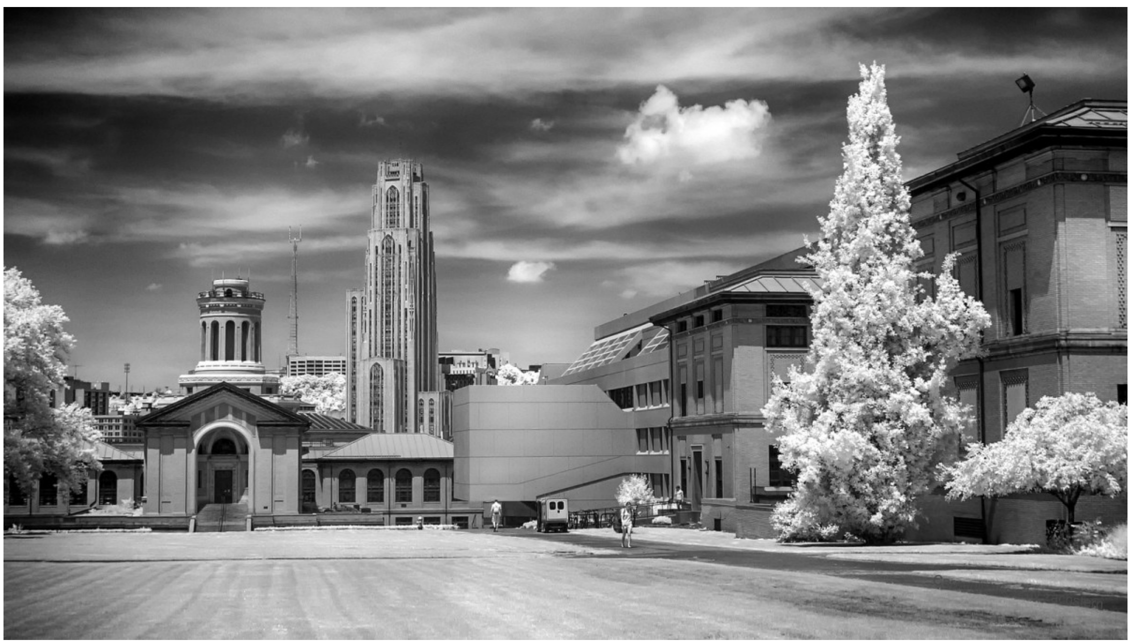
Q5

September 30, 2020

[153]:
```python
import numpy as np
from scipy import linalg
from PIL import Image
import matplotlib.pyplot as plt
```

[154]:
```python
image = Image.open('CMU_Grayscale.png')
image
```

[154]:



[155]:
```python
def svd_compress_image(compression_ratio):
    """
    Given a greyscale image of size m*n, this returns the compressed SVD␣
↪matrices Uc, sc, Vhc made from U,s,Vh such that:
    * U@S@Vh = image
    * Uc@Sc@Vhc approximately equals image
    * The total number of numbers in Uc, sc, and Vhc is as close to␣
↪compression_ratio*m*n as possible without exceeding it
```

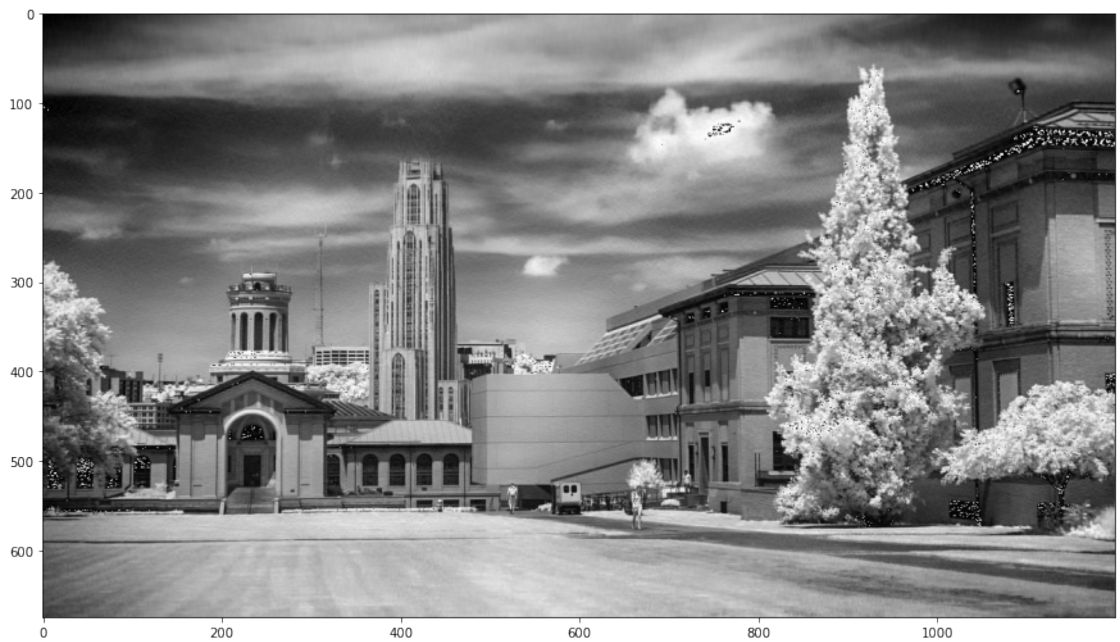
```
    Inputs:
    ----
    * compression_ratio: a number in (0,1.0] indicating the size fraction of␣
↪the original image that compressed image data should be

    Returns:
    ----
    * Uc,sc,Vhc: as described above (note: sc is an array of singular values,␣
↪not the zero-filled matrix Sc)
    * compressed_size: the total number of numbers stored in the Uc, sc, and Vhc
    * actual_compression: actual compression ratio achieved, should be <=␣
↪compression_ratio
    * compressed_image: the reconstructed post-compression image. Note: since␣
↪this will have the same dimensions as the given image, it will not be␣
↪reduced in size. That's the job of Uc, sc, and Vhc.

    Compression Theory:
    ----
    If, for an m*n image, we choose to keep ns singular value modes, we'll need␣
↪to keep track of:
    m*ns numbers in the first ns columns of U,
    ns singular values
    and n*ns numbers in the first ns rows of Vh.
    Thus, for a compression ratio of R, we want to store only only R*m*n␣
↪numbers,
    so we need to solve for the number of singular value modes to keep in the␣
↪following:
    R*m*n = m*ns + ns + n*ns = (m+1+n)*ns -> ns = floor(R*m*n/(m+n+1)).
    """
    assert compression_ratio > 0.0 and compression_ratio <= 1.0, 'Invalid␣
↪compression ratio.'
    mat = np.asarray(image) # convert to array
    (m,n) = mat.shape
    original_size = m*n # original number of numbers being tracked

    U,s,Vh = linalg.svd(mat) # perform svd
    ns = int(compression_ratio*m*n/(m+n+1)) # determine number of modes to keep␣
↪and ensure we come in slightly under target size rather than slightly over
    ns = ns if ns > 0 else 1
    # Keep only ns most important modes:
    Uc = U[:,0:ns]
    sc = s[0:ns]
    Vhc = Vh[0:ns,:]

    # Reconstruct compressed image:
    Sc = np.zeros((ns,ns))
```

```
    for i in range(ns):
        Sc[i, i] = sc[i]
    compressed_image = Image.fromarray(np.array(Uc@Sc@Vhc, dtype=np.uint8))

    compressed_size = Uc.size + sc.size + Vhc.size
    actual_compression = compressed_size/m/n
    assert actual_compression <= compression_ratio, 'Compression failed.'


    return Uc,sc,Vhc, compressed_size, actual_compression, compressed_image
```

[167]:
```
def perform_compression_trial(compression_ratio):
    """Performs SVD image compression for the target compression ratio and␣
↪returns and displays results."""
    Uc,sc,Vhc, compressed_size, actual_compression, compressed_image =␣
↪svd_compress_image(compression_ratio)
    print("\n####\nAttempted to compress image of size {:0.0f} down to {:0.1f}%.
↪ Achieved final image size of {:0.0f} with compression to {:0.1f}% shown␣
↪below.\n".format(compressed_size/actual_compression, 100*compression_ratio,␣
↪compressed_size, 100*actual_compression))
    %matplotlib inline
    plt.figure(figsize = (15,15))
    plt.imshow(np.asarray(compressed_image), cmap='gray')
    plt.imsave("24677_ps3_q5_compressed_to_{:3d}.jpg".
↪format(int(100*compression_ratio)),compressed_image, cmap='gray')
```

[173]:
```
perform_compression_trial(0.5)
```

```
####
Attempted to compress image of size 810000 down to 50.0%. Achieved final image
size of 403340 with compression to 49.8% shown below.
```




```
[172]: perform_compression_trial(0.1)
```

```
####
Attempted to compress image of size 810000 down to 10.0%. Achieved final image
size of 80668 with compression to 10.0% shown below.
```

[171]:
```
perform_compression_trial(0.05)
```

```
####
Attempted to compress image of size 810000 down to 5.0%. Achieved final image
size of 39396 with compression to 4.9% shown below.
```

[ ]:

Q6.)

Let: $\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$ $\Big|$ $A = \begin{bmatrix} -3 & 3 \\ \gamma & -4 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 1 \end{bmatrix}$

let: $P = [B \mid AB \mid A^2B]$ $\Big|$ $Q = \begin{bmatrix} C \\ \hline CA \\ \hline CA^2 \end{bmatrix}$

$\because A^2 = \begin{bmatrix} 3\gamma+9 & -21 \\ -7\gamma & 3\gamma+16 \end{bmatrix}$, $P = \left[\begin{array}{c|c|c} 1 & -3 & 3\gamma+9 \\ 0 & \gamma & -7\gamma \end{array}\right]$, $Q = \left[\begin{array}{cc} 1 & 1 \\ \hline \gamma-3 & -1 \\ \hline 9-4\gamma & 3\gamma-5 \end{array}\right]$

CONTROLLABLE $\equiv \text{rank}(P) = 2 \Rightarrow$ any $\gamma \neq 0$ ($\because$ there's no non-zero value the 0 in row2 of column 1 can be multiplied by to get the value of the first row of column 1 (1).)

$\llcorner$ if $\gamma = 0$, row2 of P is all 0 $\therefore$ rank$=1$

OBSERVABLE $\equiv \text{rank}(Q) = 2 \Rightarrow$ $\begin{cases} \gamma - 3 \neq -1, & \leftarrow \text{row 2 not a multiple of row1} \\ (3\gamma-5)\cdot\frac{1}{9-4\gamma} \neq 1 & \leftarrow \text{row 3 not a multiple of row1} \\ (\gamma-3)\cdot\frac{3\gamma-5}{-1} \neq 9-4\gamma & \leftarrow \text{row 2 not a multiple of row 3} \\ 0 \neq 9-4\gamma_0 \mid 3\gamma_0-5=0 & \leftarrow \text{row 3 is not } 0 \text{ (all other rows are obviously non-zero } \forall\gamma) \end{cases}$

$\Rightarrow \begin{cases} \gamma \neq 2, \\ \gamma \neq 2, \\ \gamma \neq 2, \gamma \neq 4 \end{cases}$

$\Rightarrow \gamma \notin \{2, 4\}$

$\therefore$

$$\boxed{\begin{array}{l} \text{6.1.) The system is controllable but not observable if } \gamma=2 \text{ or } \gamma=4. \\ \text{6.2.) The system is observable but not controllable if } \gamma=0. \end{array}}$$

Q7.)

Let: $X_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}_k$ ← all 5 LED brightnesses, x_1 is left-most LED $\therefore$

6.1.) Let: $\begin{cases} X_{k+1} = AX_k + BU_k \\ Y_k = CX_k \end{cases}$ | U_k is the signal pulse going into the leftmost LED of the queue.

$\therefore A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$, ← each LED after leftmost takes on value of previous

$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, ← signal only goes into leftmost LED

$C = I_5$ ← output equals state

$$\therefore \begin{cases} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_k \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}_k \end{cases}$$

6.2.) CONTROLLABLE: $\text{rank}(P) = 5$ | $P = [B \,|\, AB \,|\, A^2B \,|\, A^3B \,|\, A^4B]$

$\because A$ is a subdiagonal shift matrix, A^k shifts the populated diagonal down k rows, that is:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad A^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\therefore P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = I_5$$

$\therefore \text{rank}(P) = 5 \therefore$ The system is controllable. This result is intuitive because given any starting X_0, any desired state X_d could be achieved in finite time by simply shifting out the elements of X_d, one time step at a time, starting with X_{d5}. Five timesteps later, X_d will be achieved.