Assignment 10 Answer

January 2, 2024

1 Assignment 10 Answer

- There are some lecture references in this document. They are notes to myself and can be ignored.

[ ]:
```
import sympy.physics.control as spc
import scipy
from scipy.integrate import odeint
import numpy as np
import sympy as sp
import sympy.abc as abc
import matplotlib.pyplot as plt
```

1.1 Q1

- Lecture 237 S is a noncausal system
- Lecture 212 E^{At} = transition matrix

The system:

$x_1(t) = e^{3t} - 2te^{3t}$

May be generated by the system:

$A = \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix}$

with the initial conditions $(1, -2)$.

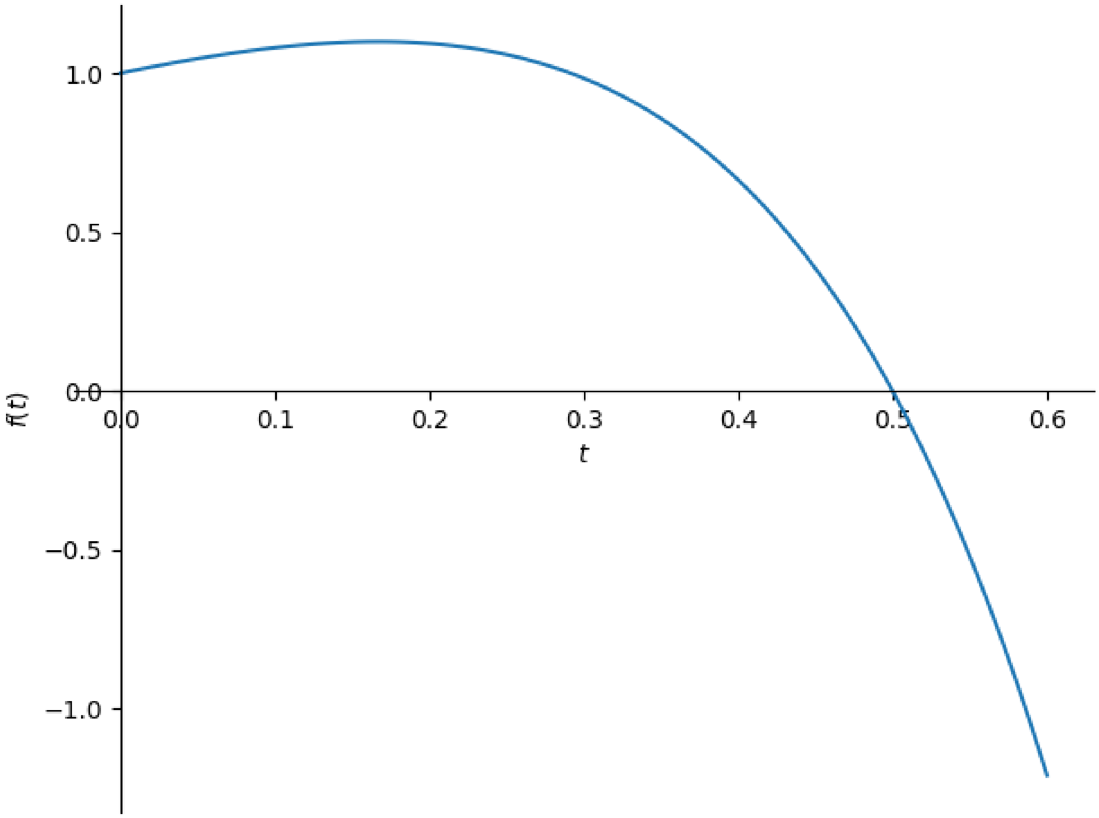
[ ]:
```
A = sp.Matrix([
    [3, 1],
    [0, 3]
])
```

[ ]:
```
t = sp.symbols('t')

y0 = sp.Matrix([1, -2])
y, dy = sp.exp(A*t)@y0
y
```

[ ]: $-2te^{3t} + e^{3t}$

[ ]:
```
sp.plot(y, (t, 0, 0.6))
plt.show()
```



1.2 Q2

Lecture 235 Partial fraction decompositions and free evolutions

- For $n > 1$ the system is not diagonisable
- That means the system does not have a full set of independent eigenvectors.

Honestly i find the Jordan representation of this quite helpful in remembering:

$$e^{At} = \begin{bmatrix} e^{\lambda} & te^{\lambda} & t^2 e^{\lambda} \\ 0 & e^{\lambda} & te^{\lambda} \\ 0 & 0 & e^{\lambda} \end{bmatrix}$$

1.3 Q3

1.3.1 1

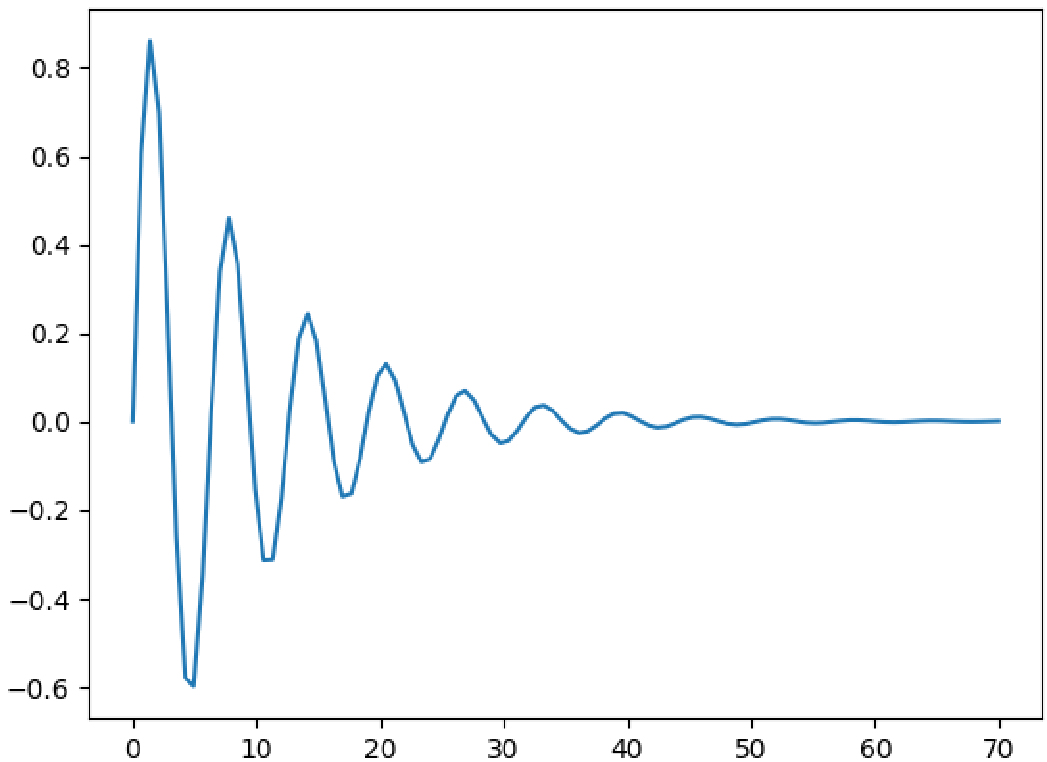
Fastly decaying oscillatory behaviour:

[ ]:
```
A, s, zeta, w = sp.symbols('A s zeta w')
```

[ ]:
```
H1 = A/(s**2 + 2*zeta*w*s + w**2)
H1
```

[ ]:
$$\frac{A}{s^2 + 2sw\zeta + w^2}$$

[ ]:
```
zeta_val = 0.1
w_val = 1
gain = 1
plt.plot(*scipy.signal.impulse(([gain], [1, 2*zeta_val*w_val, w_val**2])))
plt.show()
```



Slowly decaying exponential:

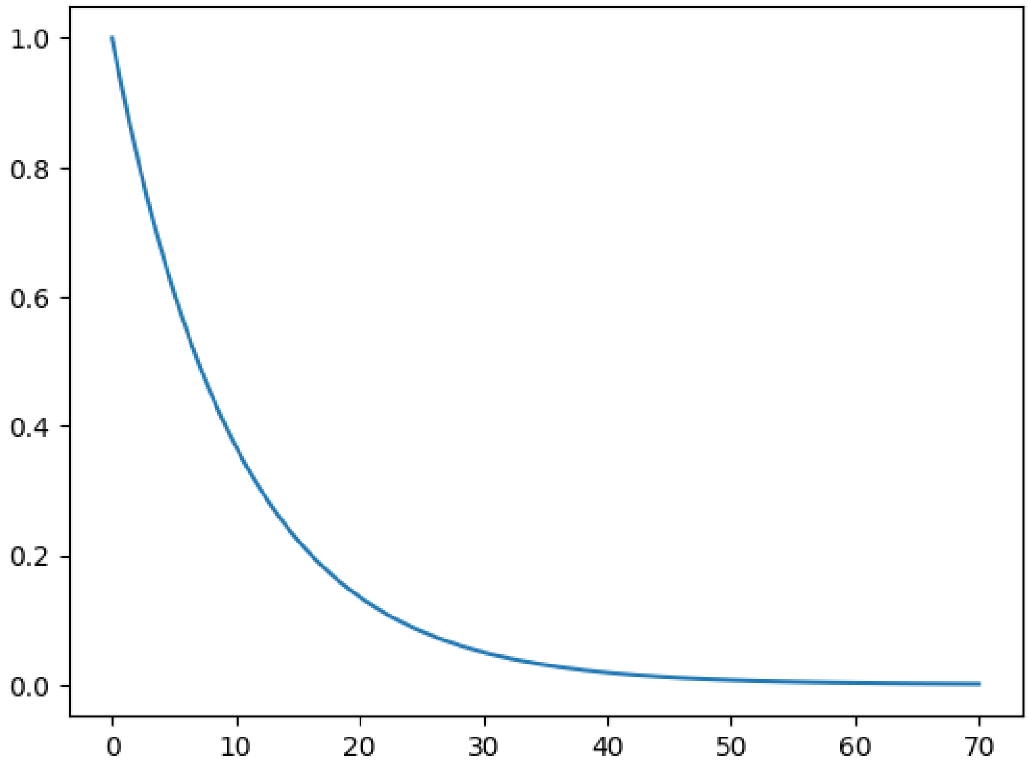
[ ]:
```
tau = sp.symbols('tau')

H2 = 1/(s + tau)
H2
```

[ ]:
$$\frac{1}{s + \tau}$$

[ ]:
```
tau_val = 0.1

plt.plot(*scipy.signal.impulse(([1], [1, tau_val])))
plt.show()
```



Combine them:

[ ]:
```
H1*H2.expand()
```

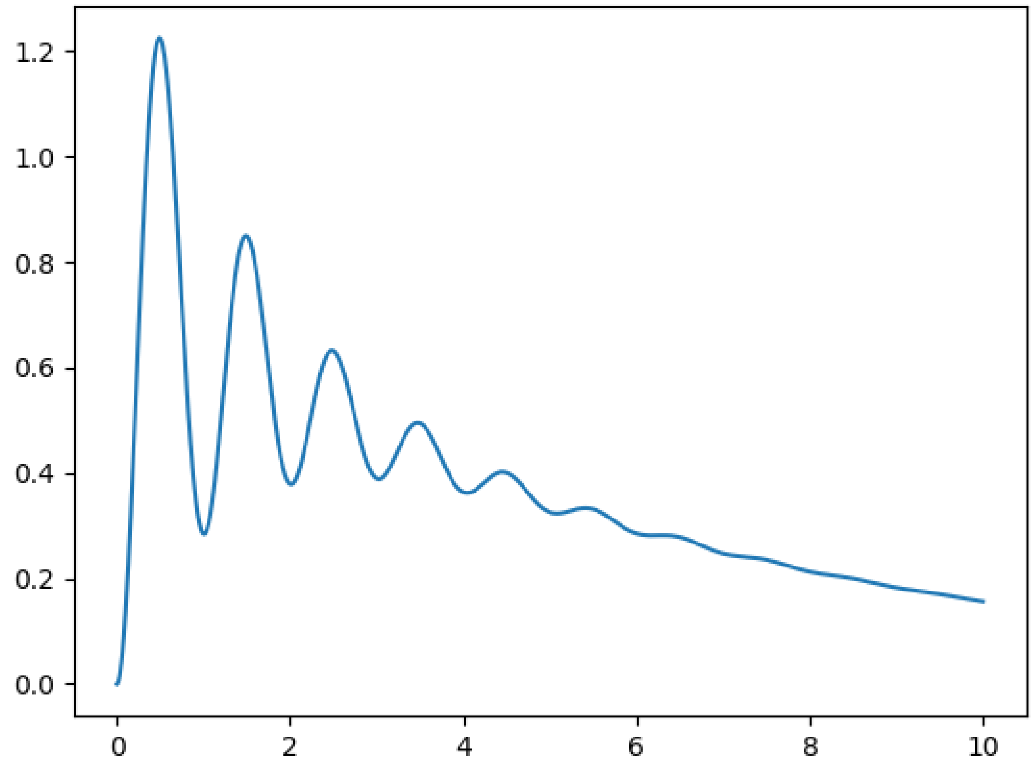
[ ]:
$$\frac{A}{(s+\tau)\left(s^2+2sw\zeta+w^2\right)}$$

[ ]:
```
(H1*H2).expand()
```

[ ]:
$$\frac{A}{s^3+s^2\tau+2s^2w\zeta+2s\tau w\zeta+sw^2+\tau w^2}$$

[ ]:
```
zeta_val = 0.05
w_val = 2*np.pi
tau_val = 1/1
gain = 30
t = np.linspace(0, 10, 1000)
```

```
plt.plot(*scipy.signal.impulse(([gain], [1, tau_val + 2*w_val*zeta_val,␣
 ↪2*tau_val*w_val*zeta_val + w_val**2, tau_val*w_val]), T=t))
plt.show()
```



The plot above shows a system with a fastly vanishing oscillatory behavior on top of a slowly vanishing exponential decay.

1.3.2 2

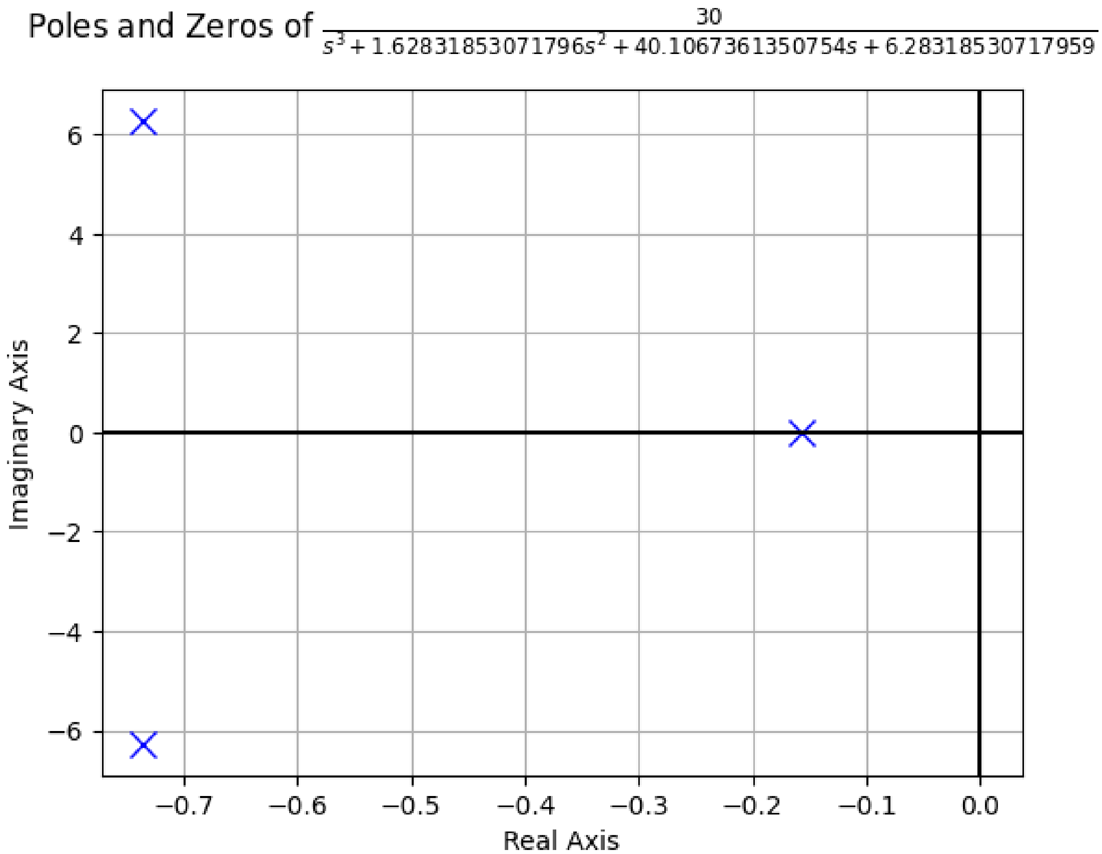
[ ]:
```
tf = spc.TransferFunction(gain, s**3 + (tau_val + 2*w_val*zeta_val)*s**2 +␣
 ↪(2*tau_val*w_val*zeta_val + w_val**2)*s + tau_val*w_val, abc.s)
tf
```

[ ]:

$$\frac{30}{s^3 + 1.62831853071796s^2 + 40.1067361350754s + 6.28318530717959}$$

The poles are the solutions for s in the polynomial of the denominator.

[ ]:
```
spc.control_plots.pole_zero_plot(tf)
```



Poles are Re < 0 and Im != 0 meaning the system is asymptotically stable.

1.3.3 3

The amplitude can be changed by increasing or decreasing the gain of the system.