Homework Two

Nathan Rose

February 10, 2022

1. A system is described by

$$\dot{x} = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 3 \\ 1 \end{bmatrix} u \tag{1}$$

Obtain the STM of the uncontrolled system using the following methods:

(a) Via taking the laplace inverse of $(sI - A)^{-1}$

$$(sI - A) = \begin{bmatrix} s+2 & -1 \\ 1 & s \end{bmatrix} \tag{2}$$

$$(sI - A)^{-1} = \begin{bmatrix} \frac{s}{s^2+2s+1} & \frac{1}{s^2+2s+1} \\ -\frac{1}{s^2+2s+1} & \frac{s+2}{s^2+2s+1} \end{bmatrix} \tag{3}$$

$$\mathscr{L}^{-1}\Big((sI - A)^{-1}\Big) = \begin{bmatrix} (1-t)\, e^{-t}\theta\,(t) & te^{-t}\theta\,(t) \\ -te^{-t}\theta\,(t) & (t+1)\, e^{-t}\theta\,(t) \end{bmatrix} \tag{4}$$

The theta in the equation is the step function as described here

This is accomplished with the code in Appendix A

(b) Via model decomposition of Matrix A
Eigenvalues are: $\lambda = -1, -1$ This is not the simple case... joy

$$(A - \lambda_i I)v_i = 0 \tag{5}$$

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} v_i = 0 \tag{6}$$

only one solution :/

$$0 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}^2 v_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} v_2 \tag{7}$$

$$v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{8}$$

$$v_1 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} v_2 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \tag{9}$$

$$V = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}, J = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} \tag{10}$$

$$\Lambda = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{11}$$

$$e^{\Lambda t} = \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-t} \end{bmatrix} \tag{12}$$

$$e^{Nt} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \tag{13}$$

$$Ve^{\Lambda t}e^{Nt}V^{-1} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}^{-1} \tag{14}$$

$$Ve^{\Lambda t}e^{Nt}V^{-1} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} e^{-t} & te^{-t} \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}^{-1} \tag{15}$$

insert some python to solve this multiplication

$$\Phi(t) = Ve^{\Lambda t}e^{Nt}V^{-1} =' \begin{bmatrix} e^{-t}(1-t) & e^{-t}t \\ -e^{-t}t & e^{-t}(t+1) \end{bmatrix}' \tag{16}$$

(c) Via the Cayley-Hamilton theorem
This will be done by using the Cayley-hamilton theorem to solve for e^{At} then via equation 22 it is the STM

- Find the characteristic polynomial

$$0 = |\lambda I - A| = \begin{vmatrix} \lambda + 2 & -1 \\ 1 & \lambda \end{vmatrix} = \lambda^2 + 2\lambda + 1 \tag{17}$$

$$\lambda = -1, -1 \tag{18}$$

- solve for β_0 and β_1

$$e^{-t} = \beta_0 - \beta_1 \tag{19}$$

$$te^{-t} = \beta_1 \tag{20}$$

$$e^{-t} = \beta_0 - te^{-t} \tag{21}$$

$$\beta_0 = e^{-t} + te^{-t} \tag{22}$$

- solve e^{At}(and STM)

$$STM = e^{At} = \beta_o I + \beta_1 A = \begin{bmatrix} e^{-t} - te^{-t} & te^{-t} \\ -te^{-t} & te^{-t} + e^{-t} \end{bmatrix} \tag{23}$$

2. In the system in Problem 1,

(a) Obtain the zero input solution $x_{Z1}(t)$ for the initial condition $\bar{x}(0) = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$

$$x(t) = STM(t)x_0 \tag{24}$$

$$x(t) = \begin{bmatrix} (10 - 9t)\, e^{-t}\theta\,(t) \\ (1 - 9t)\, e^{-t}\theta\,(t) \end{bmatrix} \tag{25}$$

(b) Obtain the zero state solution $x_{ZS}(t)$ for input $u(t) = e^{2t}$ for $t > 0$

o

$$x_{zs}(t) = \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)d\tau \tag{26}$$

$$x_{zs}(t) = \int_0^t \begin{bmatrix} (2\tau - 2t + 3)\, e^{3\tau - t}\theta\,(-\tau + t) \\ (2\tau - 2t + 1)\, e^{3\tau - t}\theta\,(-\tau + t) \end{bmatrix} d\tau \tag{27}$$

$$x_{zs}(t) = \begin{bmatrix} \frac{\left(6t + 7e^{3t} - 7\right)e^{-t}\theta(t)}{9} \\ \frac{\left(6t + e^{3t} - 1\right)e^{-t}\theta(t)}{9} \end{bmatrix} \tag{28}$$

(c) Obtain the total solution $\bar{x}(t)$ for the initial conditions and input in 2a and 2b.

Because this is an LTI system, the super-position property applies and the results of both of these systems can be added together to produce a system with initial state of 2a and the input of 2b

$$x(t) = \begin{bmatrix} \frac{\left(-75t+7e^{3t}+83\right)e^{-t}\theta(t)}{9} \\ \frac{\left(-75t+e^{3t}+8\right)e^{-t}\theta(t)}{9} \end{bmatrix} \tag{29}$$

3. Given

$$A = \begin{bmatrix} -5 & -6 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & -3 \end{bmatrix} \tag{30}$$

(a) Find A^{-1} using the Cayley-Hamilton theorem

$$\Delta(\lambda) = (\lambda I - A) = 0 \tag{31}$$

$$\lambda = -1, -2, -3 \tag{32}$$

$$f(\lambda) = \beta_0 + \beta_1\lambda + \beta_2\lambda^2 \tag{33}$$

$$\begin{bmatrix} -1 \\ -\frac{1}{2} \\ -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -2 & 4 \\ 1 & -3 & 9 \end{bmatrix}^{-1} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \tag{34}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} -\frac{11}{6} \\ -1 \\ -\frac{1}{6} \end{bmatrix} \tag{35}$$

$$A^{-1} = \begin{bmatrix} 1 & 3 & 0 \\ -1 & -\frac{5}{2} & 0 \\ 0 & 0 & -\frac{1}{3} \end{bmatrix} \tag{36}$$

(b) Obtain e^{At} using one of the three mentioned methods in Problem 1
I am using using the laplace transform method.

$$\mathscr{L}^{-1}\left((sI - A)^{-1}\right) = \begin{bmatrix} \left(4 - 3e^{t}\right)e^{-2t}\theta\left(t\right) & 6\left(1 - e^{t}\right)e^{-2t}\theta\left(t\right) & 0 \\ 2\left(e^{t} - 1\right)e^{-2t}\theta\left(t\right) & \left(4e^{t} - 3\right)e^{-2t}\theta\left(t\right) & 0 \\ 0 & 0 & e^{-3t}\theta\left(t\right) \end{bmatrix} \tag{37}$$

4. Let the STM of the system $\dot{\bar{x}}(t) = A\bar{x}(t)$, where A is a constant matrix, be $\Phi(t, t_0)$. Also, let the STM of the system $\dot{\bar{z}}(t) = -A^T\bar{Z}(t)$, where A^T is the transpose of A, be $\Theta(t, t_0)$. Use the properties of the STM on Slide 39 to show that $\Theta(t, t_0) = \Phi^T(t, t_0)$.

$$\Theta(t, t_0) = \Phi^T(t, t_0) \tag{38}$$

$$\Theta(t, t_0) = (\psi(t)\psi^{-1}(t_0))^T \tag{39}$$

$$\Theta(t, t_0) = \psi^{-1}(t_0)^T\psi(t)^T \tag{40}$$

$$\Theta(t, t_0) = (\psi(t_0)^T\psi^{-1}(t)^T)^{-1} \tag{41}$$

$$\Theta(t, t_0) = \Theta(t_0, t)^{-1} \tag{42}$$

$$\Theta(t, t_0) = \Theta(t, t_0) \tag{43}$$

5. Given a system in state space

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B\bar{u}(t) \tag{44}$$

$$\bar{y}(t) = C\bar{x}(t) + D\bar{u}(t) \tag{45}$$

prove that the transfer function matrix is invariant to any similarity transformation of the state i.e. $\bar{x} = T\bar{z}$, where T is a constant invertible matrix.

$$\dot{x} = Ax + Bu \tag{46}$$

$$T\dot{z} = ATz + Bu \tag{47}$$

$$\dot{z} = T^{-1}ATz + T^{-1}Bu \tag{48}$$

$$A_Z = T^{-1}AT, B_Z = T^{-1}B \tag{49}$$

$$C_Z = CT^{-1}, D_Z = D \tag{50}$$

$$\mathscr{L}^{-1}(C(sI - A)^{-1}B + D) \tag{51}$$

$$\mathscr{L}^{-1}(C_Z(sI - A_Z)^{-1}B_Z + D_Z) \tag{52}$$

$$\mathscr{L}^{-1}(CT^{-1}(sI - T^{-1}AT)^{-1}T^{-1}B + D) \tag{53}$$

$$\mathscr{L}^{-1}(CT^{-1}T(sI - A)^{-1}TT^{-1}B + D) \tag{54}$$

$$\mathscr{L}^{-1}(C(sI - A)^{-1}B + D) \tag{55}$$

equality!

6. Give the algebraic and geometric multiplicities of the repeated eigenvalue and find e^{3t} for the matrices below.

   (a) $J = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}$

   algebraic multiplicity: 2

   geometric multiplicity: 2

   (b) $J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$

   algebraic multiplicity: 3

   geometric multiplicity: 3

   (c) $J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$

   algebraic multiplicity: 3

   geometric multiplicity: 2

A One A source code

```python
import numpy as np
import sympy
from sympy import eye, shape, simplify, inverse_laplace_transform, Matrix
```

```
def sI_A(A: Matrix):
    s = sympy.symbols('s')
    s_I = eye(shape(A)[0])*s
    return simplify(s_I-A)

def STM_laplace_inverse(A: Matrix):
    s, t = sympy.symbols('s, t')
    return simplify(
        inverse_laplace_transform((sI_A(A)).inv(), s, t)
    )
```

B Two A source code

```
from .one_a import *
from sympy import Matrix, simplify


def zero_input_equation(A: Matrix, x_0: Matrix):
    stm = STM_laplace_inverse(A) #from problem 1
    return simplify(stm * x_0)
```

C Two B source code

```
from .one_a import *
from sympy import Matrix, simplify, exp, symbols, integrate

def get_integrand(A: Matrix, B: Matrix):
    stm = STM_laplace_inverse(A) # from problem 1
    t, tau = symbols('t, ' + r'\tau')
    u = exp(2*t)
    return simplify((stm.subs(t, t-tau) * B.subs(t, tau) * u.subs(t, tau)))

def zero_state(A: Matrix, B: Matrix):
    integrand = get_integrand(A,B)
    t, tau = symbols('t, ' + r'\tau')
    return simplify(integrate(integrand, (tau, 0, t)))
```

D Three A source code

```python
from typing import Iterable, Tuple
from sympy import Matrix, latex, eye


def beta_equation_general() -> str:
    return r"f(\lambda) = \beta_0 + \beta_1\lambda + \beta_2\lambda^2"

def beta_equation(eig_vals: list) -> Tuple[Matrix,Matrix]:
    """
    gets the matrix form of the equations above for a 3x3 and given eigenvalues.
    Assumes no multiplicity and f is inverse
    """
    values = []
    system = []
    for eig_val in eig_vals:
        values.append(
            pow(eig_val, -1)
        )
        system.append(
            [pow(eig_val, i) for i in range(len(eig_vals))]
        )
    return Matrix(values), Matrix(system)

def bmatrix() -> str:
    return r"\begin{bmatrix}\beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}"

def beta_equation_str(eig_vals) -> str:
    values, system = beta_equation(eig_vals)
    return (f"{latex(values)} = {latex(system)}" + r"^{-1}" +
            bmatrix()
            )
def solved_beta_equation_str(eig_vals) -> str:
    values, system = beta_equation(eig_vals)
    solved = system.inv() * values
    return (
        bmatrix() + f" = {latex(solved)}"
    )

def beta_values(eig_vals) -> Iterable[float]:
    values, system = beta_equation(eig_vals)
    solved = system.inv() * values
    return [solved[i, 0] for i in range(3)]

def final_answer(eig_vals, A) -> str:
    A = Matrix(A)
    I = eye(A.shape[0])
    beta_0, beta_1, beta_2 = beta_values(eig_vals)
    return (beta_0*I) + (beta_1*A) + (beta_2*A*A)
```

Disclaimer: This is just a few relevant fragments of the source code, as the entire code is a complicated system that takes these fragments and automatically renders them into the final pdf. However all of this is available online on github(its latex + python)