1. $x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\qquad \ddot{q}(t) = -(5+2\delta)\dot{q}(t) - (4+\delta)q(t)$

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(4+\delta) & -(5+2\delta) \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

$$= \left(\underbrace{\begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} 0 & 0 \\ -1 & -2 \end{bmatrix}}_{k = \begin{bmatrix} 0 \\ 1 \end{bmatrix} M = [-1, -2]} \underbrace{\delta}_{\Delta} \right) \underbrace{\begin{bmatrix} q \\ \dot{q} \end{bmatrix}}_{x}$$

$$= Ax + K\phi$$

$$\phi = \Delta \psi$$

$$\psi = Mx + \cancel{H\phi}^{\,0}$$

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} \qquad K = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$M = [-1 \;\; -2] \qquad H = 0$$

2. To find bound γ s.t. uncertain system is stable (time varying uncertain

$$\min \gamma$$

s.t. $\begin{bmatrix} PA + A^TP & PK & M^T \\ K^TP & -\frac{1}{\gamma}I & H^T \\ M & H & -\frac{1}{\gamma}I \end{bmatrix} < 0 \quad \cdots$ (SGT LMI)

$P > 0$

$$\equiv \min \bar{\gamma}$$

s.t. $\begin{bmatrix} PA + A^TP & PK & M^T \\ K^TP & -\bar{\gamma}I & H^T \\ M & H & -\bar{\gamma}I \end{bmatrix} < 0$

$P > 0$

$\downarrow$

PYTHON

$\downarrow$

$\bar{\gamma}^* = 251.6102 \Rightarrow \gamma^*$ [illegible] $\Rightarrow |\delta(t)| < \frac{1}{\bar{\gamma}^*} = \gamma^*$ [illegible]

For guarenteed stability,

3. To find bound for uncertain system with time-invarient uncertainty

i.e. if δ was time invarient,

$$A = \begin{bmatrix} 0 & 1 \\ (-4-\delta) & (-5-2\delta) \end{bmatrix}$$

Char. eq^n: $\lambda^2 - (-5-2\delta)\lambda - (-4-\delta) = 0$

$\Rightarrow \lambda^2 + (2\delta+5)\lambda - (-\delta-4) = 0$

$\Rightarrow 2\delta + 5 > 0 \Rightarrow \delta + 2.5 > 0$

or $\delta + 4 > 0 \Rightarrow \delta + 4 > 0$

$\Rightarrow \delta > -2.5$

$\Rightarrow -2.5 < \delta < 2.5$

Python Code:

```
import numpy as np
import cvxpy as cp
from scipy import signal
import control as ctrl
import matplotlib.pyplot as plt

# Define the State-Space Model of System
A = np.array([[0,    1],
              [-4,  -5]])
K = np.array([[0],
              [1]])
M = np.array([[-1, -2]])
H = np.array([[0]])

#################################################################################

# (2) Find Bounds on Disturbance for Uncertain System
# Define variables
P = cp.Variable((2, 2), symmetric=True)
gamma_bar = cp.Variable(1)
M11 = P@A + A.T@P
M12 = P@K
M13 = M.T
M21 = K.T@P
M22 = cp.multiply(-gamma_bar,np.eye(1))
M23 = H.T
M31 = M
M32 = H
M33 = cp.multiply(-gamma_bar,np.eye(1))
# LMI Problem in Small Gain Theorem (SGT)
LMI = cp.vstack([
    cp.hstack([M11[0][0], M11[0][1], M21[0][0], M31[0][0]]),
    cp.hstack([M11[1][0], M11[1][1], M21[0][1], M31[0][1]]),
    cp.hstack([M21[0][0], M21[0][1], M22[0], M23[0][0]]),
    cp.hstack([M31[0][0], M31[0][1], M32[0][0], M33[0]])
])
constraints = [LMI << 0, P >> 0]
# Set up the optimization problem
objective = cp.Minimize(gamma_bar)
problem = cp.Problem(objective, constraints)
# Solve the LMI problem
problem.solve()
```

```
# Get the value of gamma_bar (energy-to-energy gain)
gamma_bar_star = gamma_bar.value[0]
gamma_star = 1/gamma_bar_star
print(f'Optimal Solution (γ*): {gamma_star:.4f}')
delta_bounds = gamma_star
print(f'Stability Guaranteed for |δ(t)| < {delta_bounds:.4f} i.e. -
{delta_bounds:.4f} < δ(t) < {delta_bounds:.4f}')
```

Output:

Optimal Solution (γ*): 2.4173

Stability Guaranteed for |δ(t)| < 2.4173 i.e., -2.4173 < δ(t) < 2.4173

Screenshot: