EE444/544 Spring 2020 Homework #5

Due: May 3rd. You must work alone, no collaboration is permitted.
Matlab is allowed; submit your code with the solution.

Problem. Here we implement all the steps to solve the mixed sensitivity minimization problem

$$\gamma_{\text{opt}} = \inf_{C \in \mathcal{C}(P)} \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty , \quad \text{where} \quad S = (1 + PC)^{-1} , \quad T = 1 - S,$$

for the plant

$$P(s) = \frac{N(s)}{D(s)} , \quad N(s) = \frac{20\ (s - 0.5)\ (s - 1)}{(s + 10)\ (s^2 + s + 1)} , \quad D(s) = \frac{(s^2 - s + 1)}{(s^2 + s + 1)}$$

and the weights

$$W_1(s) = \frac{1 + 0.1\ s}{s + 0.1} , \quad W_2(s) = 0.1\ (s + 0.1).$$

(a) Perform inner-outer factorizations for N and D,

$$N(s) = N_i(s)N_o(s) , \quad D(s) = D_i(s)D_o(s)$$

where N_i, D_i are inner and N_o, D_o are outer.

(b) Find $X, Y \in \mathcal{H}_\infty$ such that $N(s)X(s) + D(s)Y(s) = 1$, where X is in the form

$$X(s) = \frac{x_1 s + x_0}{s + 4} , \quad x_1, x_2 \in \mathbb{R}.$$

Please pay attention to the requirement that X is first order and has a pole at -4.

(c) Find G from the following spectral factorization: $G(s)$ satisfies

$$W_1(-s)W_1(s) + W_2(-s)W_2(s) = G(-s)G(s)$$

and all the poles and zeros of $G(s)$ are in $\mathbb{C}_-$.

(d) Define $V(s) = \dfrac{W_1(s)W_2(s)}{G(s)}$ and perform the following bi-section search for γ_{opt} and C_{opt}.

Step 0. Let $\gamma_{\max} = 100$ and $\gamma_{\min} = \|V\|_\infty$

Step 1. Let $\gamma = \dfrac{(\gamma_{\max} + \gamma_{\min})}{2}$ and perform the spectral factorization

$$V_\gamma(-s)V_\gamma(s) = \gamma^2 - V(-s)V(s) , \quad V_\gamma, V_\gamma^{-1} \in \mathcal{H}_\infty.$$

Step 2. Define

$$R_\gamma(s) = V_\gamma^{-1}\Big(N_i(-s)D_i(-s)W_1(-s)G^{-1}(-s)W_1(s) - D_i(-s)G(s)N_o(s)X(s)\Big).$$

and check if there exists $Q_1 \in \mathcal{H}_\infty$ such that $\|R_\gamma - Q_1\|_\infty \leq 1$. For this purpose, solve the best achievable one-block problem performance level

$$\gamma_1 = \inf_{Q_1 \in \mathcal{H}_\infty} \|R_\gamma - Q_1\|_\infty$$

and check if $\gamma_1 \leq 1$;

if yes, then set $\gamma_{\max} = \gamma$

if no, then set $\gamma_{\min} = \gamma$

If $(\gamma_{\max} - \gamma_{\min}) \leq 0.00001$, go to Step 3; otherwise return to Step 1.

Step 3. Define $\gamma_{\text{opt}} = \gamma$ and find the optimal solution $Q_{1,\text{opt}} \in \mathcal{H}_\infty$ of the one-block problem for the latest value of γ as follows:

$$Q_{1,\text{opt}}(s) = R_\gamma(s) - F_{\text{opt}}(s) \quad \text{where} \quad F_{\text{opt}}(s) = \gamma_1 \frac{C(sI-A)^{-1}x_{\max}}{B^{\mathrm{T}}(sI+A^{\mathrm{T}})^{-1}y_{\max}},$$

with $C(sI-A)^{-1}B$ being a state space realization of $R_{\gamma,u}(s)$, the unstable part of $R_\gamma(s)$; $x_{\max}$ and $y_{\max}$ are computed as described in the class. *Pay attention to the notation: the optimal one-block performance level is γ_1, which is approximately 1.*

Step 4. Define $Q_{\text{opt}}(s) = V_\gamma(s) Q_{1,\text{opt}}(s)$, and

$$Q_c(s) = \frac{Q_{\text{opt}}(s)}{N_o(s) D_o(s) G(s)}, \qquad C_{\text{opt}}(s) = \frac{X(s) + D(s)Q_c(s)}{Y(s) - N(s)Q_c(s)}.$$

Attention: there are many approximate pole-zero cancellations in the final computations of $Q_{1,\text{opt}}$, Q_{opt} and Q_c, at the end they should all be stable.

Step 5. This is the verification step. First check that there is no unstable pole zero cancellation in the product $P(s)C_{\text{opt}}(s)$, then check that $S_{\text{opt}} = (1 + PC_{\text{opt}})^{-1}$ is stable. Alternatively, obtain the Nyquist plot and check closed loop system stability. Verify that the optimal performance level is achieved, i.e.

$$\gamma_{\text{opt}} = \left(|W_1(j\omega) S_{\text{opt}}(j\omega)|^2 + |W_2(j\omega)(1 - S_{\text{opt}}(j\omega)|^2 \right)^{-\frac{1}{2}} \quad \forall\, \omega.$$

Helpful information: You may find the spectral factorization code on p.26 of the OGKY2018 book useful. See also the Matlab code for a specific example of the Nehari problem on p.21 of the same book. If you write a generic code for arbitrary order transfer functions extending this homework, you will have your own tool for general mixed sensitivity minimization (Matlab's `mixsyn` command does the same job, but it makes some unnecessary assumptions on the plant and the weights; in fact W_2 in this problem does not satisfy `mixsyn`'s assumption).