Computer Exercises for Advanced Control Systems

Alireza Karimi
Philippe Schuchert, Vaibhav Gupta, Baiwei Guo

Spring 2022

1 CE1: Norms of Systems and Model Uncertainty

1.1 Norms of SISO systems

Consider the following second order model:

$$G(s) = \frac{10 - 2s}{s^2 + 0.1s + 16} \tag{1}$$

1.1.1 2-Norm

Compute the two-norm of G using:

1. The residue theorem (pen and paper).
2. The frequency response of G (by approximation of the integral).
3. The impulse response of G (use `impulse`).
4. The state-space method (use `are` to solve the Algebric Riccati Equation and find L).
5. Validate your results with the Matlab command `norm`.

1.1.2 ∞-Norm

Compute the infinity norm of G using:

1. The frequency response of G.
2. The bounded real lemma (iterative bisection algorithm).
3. Validate your results with the Matlab command `norm`.

1.2 Norms of MIMO systems

Download `G_mimo` that contains the state space model of a MIMO system.

1.2.1 2-Norm

Compute the two-norm using:

1. The frequency response method (by approximation of the integral).
2. The state-space method (use `are` to solve the Algebric Riccati Equation and find L).
3. Validate your results with the Matlab command `norm`.

1.2.2 ∞-Norm

Compute the infinity norm using:

1. The frequency response method.
2. The bounded real lemma (iterative bisection algorithm).
3. Validate your results with the Matlab command `norm`.

1.3 Uncertainty modeling

The objective of this part is to convert parametric uncertainty to multiplicative frequency-domain uncertainty. Consider the following model:

$$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

where $K = 10 \pm 2$, $\tau_1 = 2 \pm 0.5$ and $\tau_2 = 4 \pm 1$. Compute the weighting filter $W_2(s)$ using Matlab:

- Define the uncertain parameters K, τ_1, τ_2 using `ureal` command.
- Define the uncertain LTI system using the uncertain parameters (you can observe the step response and Bode or Nyquist diagram of the uncertain system using `step, bode, nyquist` commands).
- Use `usample` to generate two multimodel uncertainty sets: one based on 20 samples and the other one based on 200 samples.
- Use `ucover` command to convert the multimodel uncertainty to multiplicative one. Compare the two weighting filters for 20 and 200 samples.

2 CE2: Robust Control of an Electromechanical System

The plant consists of three disks supported by a torsionally flexible shaft which is suspended vertically on anti-friction ball bearings (see Fig. 1). The shaft is driven by a brushless servo motor connected via a rigid belt (negligible tensile flexibility) and pulley system with a 3:1 speed reduction ration. An encoder located on the base of the shaft measures the angular displacement of the first disk. The second disk is connected to its encoder by a rigid belt/pulley with a 1:1 speed ratio. The input of the system is the voltage of the servo motor and the output is the position of the third disk. The inertia of the disks can be changed by adding some extra masses on each disk.

For the identification purpose, the system is excited (in no extra mass case) with a PRBS signal and the acquired data are saved in `data1.mat`. The sampling period is $T_s = 0.04s$.

2.1 Multiplicative uncertainty

The objective of this part is to study the model uncertainty in the torsional system originated from the measurement noise as well as uncertainty originated from different disk loads. Finally, multimodel uncertainty is transformed to multiplicative uncertainty and the uncertainty filter is computed.

In order to study the model uncertainty from the measurement noise:

1. Download `data1.mat` that contains the input/output data of the system (current reference for the DC motor and angular position of the third disk). The data should be used to create an `iddata` object in Matlab and to identify a parametric model for the system using the *Output Error* structure. The following codes should be used to obtain the same model for all groups.

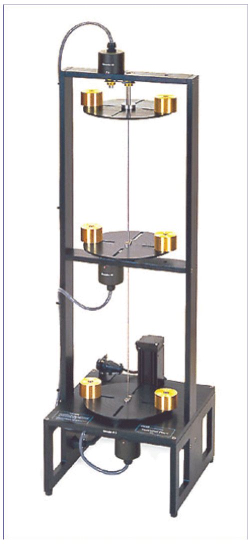

FIGURE 1 – Torsional system

```
load data1
Z=iddata(y,u,Ts);
Zd=detrend(Z);
G1 = oe(Zd,[4 6 2]);
G1f=spa(Zd,200);
```

2. For the identified model using the spectral analysis observe the frequency-domain uncertainty originated from measurement noise in the Nyquist diagram. Use the following code

```
P = nyquistoptions;
P.ConfidenceRegionDisplaySpacing=1;
P.ShowFullContour='off';
P.Xlim=[-1 1];
P.Ylim=[-0.3 1.6];
figure(1);h=nyquistplot(G1f,P);
axis equal ; showConfidence(h,2);
```

   to obtain the uncertainties with 95% confidence level.

3. Observe the frequency-domain uncertainty in the Nyquist diagram using the parametric model. Comment on the form and size of uncertainty obtained from two models.
4. With the same order and structure identify `G2` and `G3` for the data in `data2` and `data3`.
5. Choose a nominal model and compute the weighting filter W_2 that converts the multimodel uncertainty to multiplicative uncertainty.

   Hint: You can use `Gmm=stack(1,G1,G2,G3)` to define a multimodel set and `[Gu,info]=ucover(Gmm,Gnom)` to compute an uncertain model (multiplicative by default) and `W2=info.W1opt` to compute the uncertainty optimal filter (it can also be approximated with a rational fixed-order filter).
6. What is the best choice for the nominal model?

2.2 Model-based $\mathcal{H}_\infty$ control design

In this part we design a discrete-time $\mathcal{H}_\infty$ controller for the torsional system using mixed sensitivity method.

1. Design the weighting filter $W_1(z)$ for
   – Zero steady state tracking error for a step reference.
   – A modulus margin of at least 0.5.
   – The shortest settling time for the nominal model.
   Hint: You can design your weighting filter in continuous-time and then convert it to discrete-time. Check the value of the magnitude of W^{-1} in high frequencies (it should be less than 6dB). You can alternatively use the `makeweight` command.
2. Use $W_2(z)$ for multiplicative uncertainty and design an $\mathcal{H}_\infty$ controller for robust performance using the mixed sensitivity approach (use `mixsyn`).
3. Plot the step response of the closed-loop system (output and control signal), the magnitude of the input sensitivity function $U(z)$ and the sensitivity function $S(z)$.

   Hint: Use the `feedback` command as `S=feedback(1,G*K)`, `T=feedback(G*K,1)` and `U=feedback(K,G)` to compute the sensitivity functions.
4. For a unit step reference signal, the control signal $u(t)$ should be less than 2v to avoid saturation in real-time implementation. Moreover, the magnitude of $U(e^{j\omega})$ should be less than 20dB in high frequencies in order to avoid the amplification of the measurement noise on the actuator. If these conditions are not met, use $W_3 = cte.$ to reduce the magnitude of $U(e^{j\omega})$ and consequently the magnitude of $u(t)$.
5. The order of the final controller may be too large (especially if the order of $W_2(z)$ is large). Check if there is zero/pole cancellation in the controller using `pzmap`. The order of the controller can be reduced using the `reduce` or `minreal` command. Check the stability and performance of the closed-loop system with the reduced order controller.

2.3 Model-Based $\mathcal{H}_2$ Controller Design

In this part an $\mathcal{H}_2$ state feedback controller is designed for the model `G3` of the active suspension system. The objective is to design a state feedback controller such that the sum of the two-norm of the closed loop transfer functions from the input disturbance to the output and to the input of the system is minimized.

1. Convert the discrete-time model `G3` to a continuous-time model (use `d2c`).
2. Write the state space equations of the closed-loop system (see Chapter 2, slide 20). In order to minimize the sum of two transfer functions we define $y_1(t) = Cx(t)$ and $y_2(t) = -Kx(t)$.
3. Write a convex optimization problem using LMIs that represents this problem.
4. Compute the controller K by solving the SDP problem using YALMIP.
5. Plot the step response of the closed-loop system.
6. Compare your result with the Linear Quadratic Regulator computed using `lqr` command of Matlab.