Practice 1 - Study of a multivariable system

Prof. Lalo Magni, Prof. Chiara Toffanin

Abstract

The main objective of this practice is to study the multivariable systems.

Exercises

1. Calculate poles and zeros of the multivariable system described by the transfer function (1):

$$G(s) = \begin{bmatrix} \frac{(s+1)}{(s+2)(s+3)} & \frac{4}{(s+2)(s+3)} \\ \frac{0.5}{(s+2)(s+3)} & \frac{2}{(s+2)(s+3)} \end{bmatrix} \tag{1}$$

   Is the system asymptotically stable?

2. Calculates poles and zeros for each single transfer function of (1).

3. By using the Simulink schema **frequence_response.mdl** shown in Figure 1, simulate the system (1) with simulation time equal to $30s$.

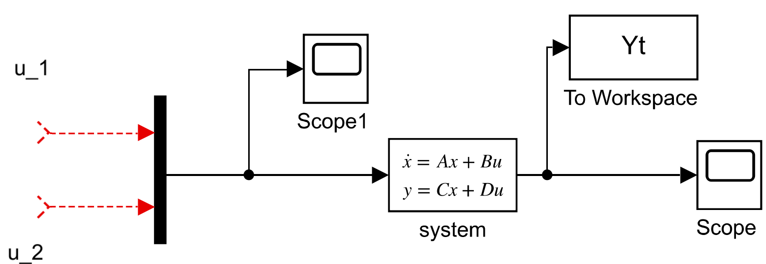

Figure 1: MIMO system described by $G(s)$

   The first input (u_1) is a sinusoid with amplitude 3 and frequency 2 rad/sec while the second one (u_2) is a sinusoid with amplitude 6 and frequency 2 rad/sec. Verify that $\frac{||Y(j\omega)||_2}{||U(j\omega)||_2}$ resides between the minimum and maximum singular values of the system of (1).

4. Execute a new simulation with simulation time equal to $30s$. The first input (u_1) is a constant with amplitude 4 while the second one (u_2) is a sinusoid with amplitude 6 and frequency 2 rad/sec.

Verify that $\frac{||Y(j\omega)||_2}{||U(j\omega)||_2}$ resides between the minimum and maximum singular values of the system (1). Why both the outputs are sinusoidal?

5. Execute a new simulation with simulation time equal to $30s$. The first input (u_1) is a constant with amplitude 1 while the second input (u_2) is a constant with amplitude -0.25. Verify that $\frac{||Y(j\omega)||_2}{||U(j\omega)||_2}$ resides between the minimum and maximum singular values of the system (1). Why both the outputs are zero?

6. Consider a regulator $R(s) = I$. Is the closed-loop system asymptotically stable? (Advice: use the "Small Gain Theorem").

7. Consider a controller $R(s)$ in which the diagonal elements are two *integrators* $\left(\frac{1}{s}\right)$. Is it possible to use the schema with the integrators assuring the closed-loop stability?
Calculate poles and zeros of the closed-loop transfer function.

8. By using the Simulink schema **frequence_response_integrators.mdl**, shown in Figure 2, simulate the system (1) controlled by the regulator $R(s)$ of the preceding point with simulation time equal to $80s$.

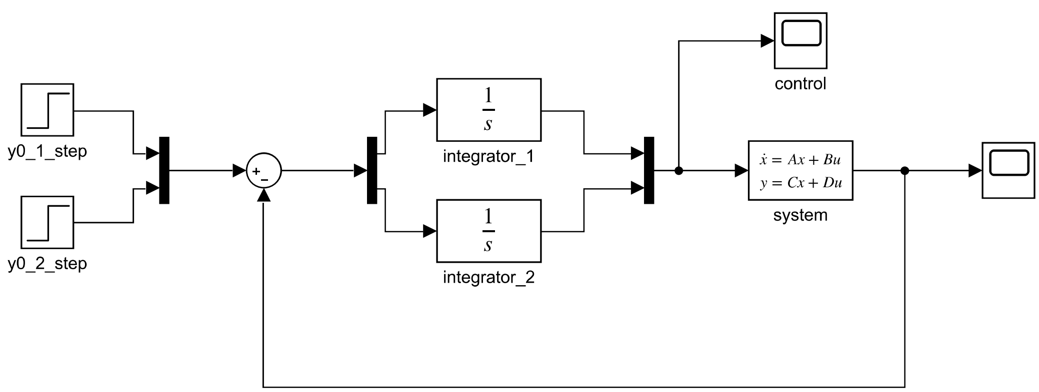

Figure 2: Feedback system with regulator including integrators

The first set point (y_{01}) is a constant with amplitude 4 starting from $t = 2s$ while the second one (y_{02}) is a constant with amplitude 6 starting from $t = 10s$. By referring to the stability study of the preceding point, why both the system outputs are stable?

9. Change the output transformation of the LTI system (1) with:

$$y(t) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x(t). \tag{2}$$

Calculates poles and zeros of the new system.

10. Repeat the simulation required at point (8) with the system defined at point (9). Verify that perfect tracking of references is now ensured.

Matlab References

- **tf([num],[den])**: creates a transfer function with the coefficients of the numerator and denominator polynomials specified in *num* and *den* vectors, respectively.
- **s = tf('s')**: defines the s laplace variable in Matlab workspace.
- **sys=ss(A,B,C,D)**: creates a LTI system starting from the state-space matrices A,B,C,D.
- **[A, B, C, D] = ssdata(sys)**: given a LTI system *sys* (or transfer function), returns the state spaces representation matrices A, B, C, D.
- **trim**: calculates an equilibrium point of a Simulink model.
- **linmod**: linearize a Simulink model around a specified equilibrium point.
- **sigma(sys)**: plot of the minimum and maximum singular values of the system *sys*.
- **norm(A,p)**: calculates the *p-norm* of the A matrix (p can be equal to *1*, *2* or *inf*).
- **eig(M)**: calculates the eigenvalues of the M matrix.
- **rank(M)**: calculates the rank of the M matrix.
- **ctrb(sys)**: calculates the controllability matrix of the linear system *sys*.
- **obsv(sys)**: calculates the observability matrix of the linear system *sys*.
- **pzmap(sys)**: plot of the poles and zeros of the linear system *sys* in the complex plane.
- **tzero(sys)**: calculates zeros of the linear system *sys*.
- **pole(sys)**: calculates poles of the linear system *sys*.
- **To Workspace**: sinks library.
- **Mux**: signal routing library.