Computer controlled systems

Dynamical systems simulations in Matlab/Simulink

version: 2019.11.15. – 08:42:10

1 RLC circuit

The dynamics of the RLC circuit can be given by the following linear model:

$$\begin{cases} \dot{x}_1 = -\dfrac{R}{L}x_1 - \dfrac{1}{L}x_2 + \dfrac{1}{L}u & \text{(state equations)} \\ \dot{x}_2 = \dfrac{1}{C}x_1 & \\ y = x_2 & \text{(output equation)} \end{cases} \tag{1}$$

where $x_1 = i$, $x_2 = u_C$, $u = u_{\text{in}}$ is the input, $y = x_2 = u_C$ is the output.

1. Simulate the system using `ode45`, with no input.
2. Compute the $H(s)$ transfer function of the system.
3. Compute impulse and step response of the system.
4. Simulate the system in Simulink with a given input.

2 Lotka Volterra model (simple ecological system)

In this section, I give two different predator-pray dynamics:

$$\Sigma_1 : \begin{cases} \dot{x}_1 = x_1(-x_1 + 2) \\ \dot{x}_2 = x_2(0.2x_1 - 1) \end{cases} \qquad \Sigma_2 : \begin{cases} \dot{x}_1 = x_1(-2x_1 - 3x_2 + 5) \\ \dot{x}_2 = x_2(1.4x_1 + x_2 - 2.4) \end{cases} \tag{2}$$

1. Compute the equilibrium points of this nonlinear system. You can use `fsolve`, which is function of the Symbolic Math Toolbox (SMT).
2. Determine which equilibrium points are stable.
3. Simulate the system with different initial conditions around the stable equilibrium point.

3 Continuous fermentation process

The dynamic equation of the continuous fermentation (or bioreactor) process is the following:

$$\begin{cases} \dot{x}_1 = \mu(x_2)x_1 - \dfrac{x_1 F}{V} \\ \dot{x}_2 = -\dfrac{\mu(x_2)x_1}{Y} + \dfrac{(S_F - x_2)F}{V} \end{cases} \quad \text{where} \quad \mu(x_2) = \frac{x_2 \mu_{\max}}{K_2 x_2^2 + x_2 + K_1} \tag{3}$$

The value of the constants F, V, Y, S_F, K_1, K_2, $\mu_{\max}$ are given in the Matlab script.

1. Compute the equilibrium points of this nonlinear system. You can use `fsolve`, which is function of the Symbolic Math Toolbox (SMT).
2. Determine which equilibrium points are stable.
3. Simulate the system with different initial conditions around the stable equilibrium point.
4. Define a proper terminal condition for `ode45`:
   - Terminate if the solution is far enough from the equilibrium point.
   - Terminate if any of the state variables reaches zero.
5. Color the convergent and divergent trajectories with two different colors (eg. red if convergent, blue if divergent).
6. Simulate the dynamics in Simulink.