Ecole Centrale de Nantes
Project

Moein SARBANDI

July 5, 2026

SIMUS

Goal

The objective is to implement the basic tools of **Matlab/Simulink** in order to simulate a state system described by nonlinear equations, and its associated control.

1 Control and differentiation

We consider the following system

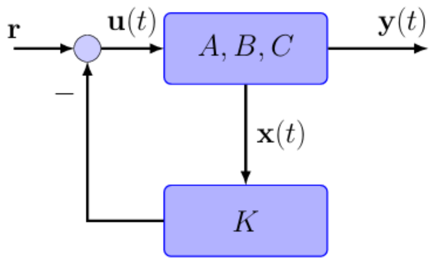
$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -x_1 \cdot x_2 + x_3^2 + u \end{aligned} \tag{1}$$

x_1, x_2, and x_3 being the state variables and u being the control input. The control is nonlinear in nature and defined by

$$u = x_1 \cdot x_2 - x_3^2 + v, \tag{2}$$

the term v being to be defined.

1. Show that the control u makes it possible to the system (1) be linear.
Then propose a control law v allowing x_1 to follow a sinusoidal trajectory with angular frequency 1 rad/s and amplitude 2. This control v will be based on linear state feedback (for tuning, use `acker` – multiple pole at -2).



2. Simulate the system (initial condition $[x_1(0)\ x_2(0)\ x_3(0)]^T = [0\ 0.9\ 0.1]^T$) and the control using Matlab functions, and choosing a fixed-step integration algorithm. Compare the simulation results with `Euler` and `Runge Kutta`, for increasingly larger sampling periods. **First, we assume that x_1, x_2, and x_3 are measured.** Plot the state variables on one figure (use `subplot`) and the control input u on another figure.

3. We now use the `Runge Kutta` integration algorithm with $\tau = 0.1\ ms$. Suppose now that only the variable x_1 is measured, with variables x_2 and x_3 being estimated using two differentiators filtered by a first-order system

$$\frac{s}{1+\tau_e s}$$

To evaluate the performance of this filter, first simulate the filtered differentiators in "open loop" (that is, the derivative estimates are not fed into the control law). Analyze the quality of the estimation as a function of the value of τ_e. Plot on the same figure x_2 and its estimate on one hand, and x_3 and its estimate on the other hand. Comment.
By setting $\tau_e = 1\ ms$, place the filtered estimates in the control loop, and plot x_1 and u.

4. Another technique for evaluating unmeasured variables is the use of an observer. From the system (1) to which we apply the control (2) (which yields a linear system), compute a **Luenberger**-type observer (multiple pole at -2). Using a *Matlab Function* block, integrate the observer into the simulator. Remember to initialize the observer with initial values different from those of the system ($[0,\ 1.1,\ 0.2]^T$). First test the observer not embedded in the control loop (plot the estimation errors), then in a second step, test the observer in the control loop. Verify that the estimation errors for all the state variables and the tracking error for x_1 converge to 0.

2 Dynamic control

We consider the nonlinear system

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)u_1 \\ \dot{x}_2 &= \sin(x_3)u_1 \\ \dot{x}_3 &= u_2 \end{aligned} \qquad (3)$$

This system describes the dynamic behavior of a mobile robot in a plane, with x_1 and x_2 being the Cartesian coordinates. The objective is to control the two variables x_1 and x_2. To achieve this, a nonlinear dynamic control is implemented and defined by

$$\begin{bmatrix} \dot{u}_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos(x_3) & -\sin(x_3)u_1 \\ \sin(x_3) & \cos(x_3)u_1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad (4)$$

Compute the two inputs v_1 and v_2 given that they aim to stabilize the following two subsystems

$$\begin{aligned} \ddot{x}_1 &= v_1 = -k_{11}\dot{x}_1 - k_{10}x_1 \\ \ddot{x}_2 &= v_2 = -k_{11}\dot{x}_2 - k_{20}x_2 \end{aligned} \qquad (5)$$

Set up a simulator to stabilize the system at the origin. Plot the system trajectory in the plane (x_1, x_2), as well as the control inputs u_1 and u_2.

3 Observer–Controller Duality

1. Consider the linear state–space system

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t).$$

1. Show that if the control law is chosen as

$$u(t) = -Kx(t) + r(t),$$

   then the closed-loop dynamics of the system are governed by the matrix

$$A - BK.$$

2. Show that if we design a Luenberger observer of the form

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\big(y(t) - C\hat{x}(t)\big),$$

   then the estimation error dynamics are governed by the matrix

$$A - LC.$$

3. Explain briefly the role of the eigenvalues of $A - BK$ and $A - LC$.

4 Classical PI/PID Control

5. Consider the following system

$$G(s) = \frac{1}{s(s+1)}.$$

1. Manually design the gains of a PI/PID controller for this.
2. Use MATLAB's `pidTuner` to design a PID controller for $G(s)$, and compare the closed-loop performance with the manually designed controller. (Hint: you may use the syntax `pidTuner(G,'PID')` to launch the interactive tuning tool.)
3. Plot and compare the step responses of both controllers using MATLAB. Comment on the differences in terms of overshoot, settling time, and steady-state error. (Hint: use `step(sys)` and `stepinfo(sys)` to analyze responses.)

5 Comparison of Control Methods

7. Consider a linear system in state-space form via feedback liniearization used in q1.

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t).$$

Design three different controllers for this system:

(a) A PID controller using `pidTuner`. (Hint: `pidTuner(G,'PID')`)

(b) A full-state feedback controller with pole placement, using the syntax

$$K = \texttt{acker(A,B,poles)} \quad \textit{or} \quad K = \texttt{place(A,B,poles)}.$$

(c) An optimal controller using LQR, with the syntax

$$K = \texttt{lqr(A,B,Q,R)},$$

where Q and R are user-chosen weighting matrices.

Simulate and compare the closed-loop responses of the three approaches under the same initial conditions.
Comment on the advantages and limitations of each method.