

ELP7100: Autonomous Systems & Control Lab

Introduction to Scientific Computing Softwares

Course Instructors: Subashish Datta subashish@ee.iitd.ac.in

Shaunak Sen shaunak@ee.iitd.ac.in

Course TAs: Shuvam Routray, Khushall Gourav, Anvisha Chaturvedi

1. Similarity Transformation (Controller Canonical Form):

Consider a system:

$$\dot{x} = Ax + Bu$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control input. Assume that the pair (A, B) is controllable. Transform this system to an equivalent one:

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u$$

such that $\bar{A} = PAP^{-1}$, $\bar{B} = PB$ are in *controller canonical* form.

Steps to obtain $\bar{A}$ & $\bar{B}$:

(a) Compute the controllability matrix:

$$\begin{aligned}\mathcal{C} &= [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B] \\ &= [b_1 \ b_2 \ \cdots \ b_m \mid Ab_1 \ Ab_2 \ \cdots \ Ab_m \mid \cdots \mid A^{n-1}b_1 \ A^{n-1}b_2 \ \cdots \ A^{n-1}b_m]\end{aligned}$$

(b) Starting from the left and moving towards the right, select n independent columns of $\mathcal{C}$. This is always possible since the pair (A, B) is controllable.

(c) Reorder these columns by taking first $b_1, Ab_1, \dots$ until all columns involving b_1 have been used. Then take $b_2, Ab_2, \dots$ till all columns involving b_2 are used and then lastly use $b_m, Ab_m, \dots$

(d) Construct the following matrix:

$$\mathcal{C}' = [b_1 \ Ab_1 \ \cdots \ A^{\mu_1-1}b_1 \mid b_2 \ Ab_2 \ \cdots \ A^{\mu_2-1}b_2 \mid \cdots \mid b_m \ Ab_m \ \cdots \ A^{\mu_m-1}b_m]$$

The positive integers μ_i are called the *controllability indices* of the system. Further,

$$\mu := \max_i \mu_i$$

is called the *controllability index* of the system.

(e) Define

$$\sigma_k = \sum_{i=1}^k \mu_i \quad k = 1, 2, \dots, m$$

i.e.

$$\sigma_1 = \mu_1$$

$$\sigma_2 = \mu_1 + \mu_2$$

$$\vdots$$

$$\sigma_m = \mu_1 + \mu_2 + \cdots + \mu_m = n$$

(f) Compute $[\mathcal{C}']^{-1}$, and denote q_k as the σ_k^{th} row of $[\mathcal{C}']^{-1}$.

(g) Construct the transformation matrix P as:

$$\begin{bmatrix} \dots & q_1 & \dots \\ \dots & q_1 A & \dots \\ \vdots & \vdots & \vdots \\ \dots & q_1 A^{\mu_1-1} & \dots \\ \hline \dots & q_2 & \dots \\ \dots & q_2 A & \dots \\ \vdots & \vdots & \vdots \\ \dots & q_2 A^{\mu_2-1} & \dots \\ \hline \vdots & \vdots & \vdots \\ \dots & q_m & \dots \\ \dots & q_m A & \dots \\ \vdots & \vdots & \vdots \\ \dots & q_m A^{\mu_m-1} & \dots \end{bmatrix}$$

(h) Finally, compute:

$$\bar{A} = PAP^{-1}, \quad \bar{B} = PB$$

Programming Task: Write code in **MATLAB**, **Scilab**, and **Python** to compute the controller canonical form. Your code should work for any (A, B) controllable pair for any n and m . You should not write this code for a particular example. Observe the structure of $\bar{A}$ and $\bar{B}$.

- **Input:** Matrices A, B
- **Output:**
 - (a) Canonical matrices $\bar{A}, \bar{B}$
 - (b) Controllability indices $\mu_1, \dots, \mu_m$

2. Feedback Control Design

Given a controllable system:

$$\dot{x} = Ax + Bu, \quad x(0) = x_0, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}, \quad (m = 1)$$

Choose any system of size $n \geq 4$, $m = 1$. First, convert the system into controller canonical form: $\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u$ using the previous algorithm (Problem 1). Then, apply state feedback control:

$$u = \bar{F}\bar{x}$$

where $\bar{F} \in \mathbb{R}^{1 \times n}$ is the feedback gain vector. The closed-loop system becomes:

$$\dot{\bar{x}} = (\bar{A} + \bar{B}\bar{F})\bar{x}$$

Design Objective:

- Choose $\bar{F}$ such that the eigenvalues of $(\bar{A} + \bar{B}\bar{F})$ satisfy:
 - (a) Damping ratio $\zeta \geq 0.7$,
 - (b) Decay rate $\lambda \leq 1$.
- Compute $F = \bar{F}P$

- Compute the eigenvalues of $(A + BF)$, and then compare them with the eigenvalues $(\bar{A} + \bar{B}\bar{F})$. Comments on your observations.
- Simulate the closed-loop system for an initial condition $x_0 \in \mathbb{R}^n$.

Simulation Task: Implement the controller and simulate the closed-loop response in:

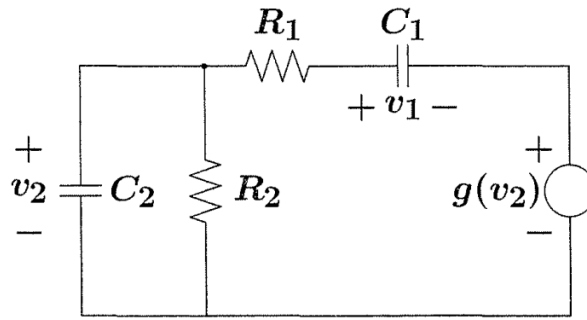
- **MATLAB**
- **Python**
- **Scilab**

Submit:

- Code file (.m, .py, .sce)
- Plot showing system response (state trajectories over time)
- Choice of closed-loop poles and justification.

3. Numerical Solution of Non-linear Systems

An equivalent circuit of the Wien-Bridge oscillator is shown below:



The non-linear voltage controlled voltage source is given by:

$$g(v_2) = 3.23v_2 - 2.195v_2^3 + 0.666v_2^5$$

Let $x_1 = v_1$ and $x_2 = v_2$. Then, a state-space model of the circuit is:

$$\begin{aligned}\dot{x}_1 &= \frac{1}{C_1 R_1} (-x_1 + x_2 - g(x_2)) \\ \dot{x}_2 &= -\frac{1}{C_2 R_1} (-x_1 + x_2 - g(x_2)) - \frac{1}{C_2 R_2} x_2\end{aligned}$$

Assume that: $C_1 = C_2 = R_1 = R_2 = 1$. Then:

- Solve the above set of equations to obtain x_1 and x_2 using *numerical integration* methods. Do not use any existing ODE solver command. Let the solutions be $\bar{x}_1$ and $\bar{x}_2$.
- Solve these equations for x_1 and x_2 using existing commands (e.g., built-in ODE solvers). Plot x_1 and x_2 . Also plot $\bar{x}_1$ and $\bar{x}_2$ (obtained from part (i)). Compare these plots.

Programming Task: For both parts (i) and (ii), use **MATLAB**, **Scilab**, and **Python**.

[Example taken from Book: Nonlinear Systems, by Khalil, Chapter 2]

4. Modeling and Simulation with OpenModelica:

Using **OpenModelica**, perform modeling and simulation of the following systems:

- Simple electric circuits
- Mass-Spring Damper system
- An electro-mechanical system

Analysis Task: Comment on the type of model you have obtained in comparison to the ordinary state-space model.

5. Consider the following model of a simple pendulum $\ddot{\theta} + \sin \theta = 0$. Define state variables as $x_1 = \theta$ and $x_2 = \dot{\theta}$. Rewrite the pendulum model as a set of two first-order differential equations.

We will explore numerical integration of these equations. The simplest method is the Euler method that approximates the derivative as $\dot{x}_1 \approx (x_1(t+h) - x_1(t))/h$, where h is a small number. Using this approximation and a similar one for $\dot{x}_2$, rewrite the two first-order differential equations as corresponding difference equations.

The two first-order difference equations can be numerically solved in a computer for any initial condition $x_1(0)$ and $x_2(0)$ and a finite time interval $[0, T]$. Divide the time interval in T/h steps of h . Denote $x_i(nh) \equiv x_n$ for $i = 1, 2$ and $n = 0, \dots, T/h$. Calculate the approximate trajectory for $x_1(0) = 1, x_2(0) = 0, T = 10, h = 0.1, 0.01, 0.001$. What are your observations for different values of h ? Compare this with a standard ODE solver in MATLAB or Python.

6. Many control applications are to heat transfer. The heat equation in one dimension is

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}. \quad (1)$$

For the x domain $0 \leq x \leq 1$, time interval $0 \leq t \leq 1$, the initial condition $T(x, 0) = 37$ and boundary conditions $T(0, t) = 25 = T(1, t)$, we wish to obtain the spatiotemporal profile of the temperature T .

Consider a discretization

$$\frac{1}{k}(T_{i,j+1} - T_{i,j}) = \frac{1}{h^2}(T_{i+1,j} - 2T_{i,j} + T_{i-1,j}), \quad (2)$$

where h is the step size in the x -direction, k is the step size in the t -direction, and $T(ih, jk) \equiv T_{i,j}$. Solve for the temperature profile for different values of h and k . What are your observations?

7. If one puts a bacterial cell in a medium, it grows and divides. This process of growth continues until the resources are exhausted. Suppose we have measurements of the growth (O_k) at discrete instants of time (t_k , attached data file). The broad question is to estimate the growth rate.

(a) Calculate the growth rate by taking a discrete approximation $\gamma_k = (O_{k+1} - O_k)/(t_{k+1} - t_k)$.

(b) Treat the growth rate as a parameter to be estimated from a model $\dot{N} = \gamma(t)N$, where $N(t)$ is the number of cells at time t . Use an Extended Kalman Filter, to estimate γ from output measurements $y(t_k) = N(t_k)$ (second column in Excel File).

The “Extension” in the Kalman Filter arises through a redefinition of the parameter γ as a state. So the two states are N and γ and the corresponding equations are $\dot{N} = \gamma N, \dot{\gamma} = 0$. The equation may be linearized around a point (N^*, γ^*) as $\Delta N = N^* \Delta \gamma + \gamma^* \Delta N, \Delta \gamma = 0$, where $\Delta N = N - N^*$ and $\Delta \gamma = \gamma - \gamma^*$.

This way the equation is in a standard linear form $\dot{x} = Ax$ and $y(t_k) = Cx(t_k)$. The Kalman Filter for this is:

- *Between Observations* ($t_k < t < t_{k+1}$)

$$\dot{\hat{x}}_t^f = A x_t^f, \quad (3)$$

$$\dot{P}_t^f = A P_t^f + P_t^f A' + Q. \quad (4)$$

- *At an Observation* ($t = t_k$)

$$\dot{\hat{x}}_{t_k}^{f+} = x_{t_k}^{f-} + K(t_k)(y(t_k) - C x_{t_k}^{f-}), \quad (5)$$

$$\dot{P}_{t_k}^{f+} = P_{t_k}^{f-} - K(t_k) C P_{t_k}^{f-}. \quad (6)$$

Here, the $\hat{x}$'s represent the state estimates, P 's represent the estimate covariances:

$$K(t_k) = P_{t_k}^{f-} C' (C P_{t_k}^{f-} C' + R)^{-1}$$

is the Kalman Gain, $Q \in R^{2 \times 1}$ is the process noise covariance, and $R \in R^{1 \times 1}$ is the measurement noise covariance.

(c) Compare the growth rate estimated in (a) and (b) above.

