Indian Institute of Technology, Bombay — Chemical Engineering

CL686: Advanced Process Control — **Assignment 3, 2025**

Date Due: 09 March 2025, 11.59 PM (Note that time is not 12 PM midnight since that is confusing to set.)

Mode of Submission: Moodle submission

Late Submission Policy: Submissions late by 0-24 hrs will incur a 25% penalty, and submissions late by 24-48 hrs will incur a 50% penalty. Submissions later than 48 hrs will not be considered.

Topic: Pole Placement Controller and LQOC controller design

AIM: To implement Pole Placement Controller and LQOC controller for Quadruple Tank System.

Task at Hand

1. Implement Pole Placement Controller (PPC).
2. Implement Linear Quadratic Optimal Controller (LQOC) using a steady-state solution of the Riccati equation.
3. Implement Linear Quadratic Optimal Controller (LQOC) by solving the Riccati equations backwards in time.
4. Compare the performance of the above controllers with the Model predictive controller (MPC) in terms of Sum of squared error (SSE) value. (MPC was implemented in Assignment 2).
   NOTE: All controllers must be implemented considering nonlinear ODEs of Quadruple Tank System as the plant. The controllers will be based on linear-model though.

Quadruple Tank System Description

Consider a nonlinear system governed by the following set of coupled nonlinear ODEs,

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1 \tag{1}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2 \tag{2}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2 + \frac{\gamma_3}{A_3}F_d \tag{3}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1 + \frac{(1-\gamma_3)}{A_4}F_d \tag{4}$$

The initial steady-state operating conditions of the system are listed in the table below:

Variable	Optimum operating point
A_1, A_3 $[cm^2]$	28
A_2, A_4 $[cm^2]$	32
a_1, a_3 $[cm^2]$	0.071
a_2, a_4 $[cm^2]$	0.057
g $[cm/s^2]$	981
$k1, k2$ $[cm^3/Vs]$	3.14, 3.29
$\gamma_1, \gamma_2, \gamma_3$	0.7, 0.6 , 0.4

Note: In this assignment, we consider a minimum phase scenario (i.e., values of γ_1 and γ_2 greater than 0.5) to ensure that PPC and LQOC can effectively track the setpoint.

- **System State:** h_1, h_2, h_3, h_4 are the liquid level in the quadruple tank system.
- **Process inputs:** v_1 and v_2 are the voltage applied to the pump.
- **Disturbance inputs:** F_d act as the disturbance input to the process. For this assignment, it will not be considered (i.e. its value will be kept at 0). [However, enterprising students can investigate its effect as an additional not-to-be-submitted exercise]

Condition for Simulation:

- **Steady state inputs:** $U_s = [3 \quad 3]^T$; $D_s = 0$
- **Equilibrium/Steady State Operating Point:**
  $X_s = [12.2630 \quad 12.7831 \quad 1.6339 \quad 1.409]^T$

- **Sampling interval**, $T = 4$ sec
- **Number of samples**, $N_s = 200$
- **Initial state**, $X(0) = X_s + [2 \quad 2 \quad 2 \quad 2]^T$.
- **Setpoint**, $R = X_s$
- **Input bound constraints**, $U_L = [0 \quad 0]^T$, $U_H = [5 \quad 5]^T$
- **Input rate constraints**, $\Delta U_L = [-3 \quad -3]^T$, $\Delta U_H = [3 \ \ 3]^T$
- **State bound constraints**, $X_L = [10 \ \ 10 \ \ 2 \ \ 2]^T$, $X_H = [16 \ \ 16 \ \ 8 \ \ 8]^T$

POLE PLACEMENT CONTROLLER IMPLEMENTATION

- **Desired controller pole:** $p = [0.9382 \quad 0.9567 \quad 0.8458 \quad 0.8751]^T$
- **Control gain matrix (G):** Controller gain matrix (**G**) can be obtained using MATLAB inbuilt command *place* as follows
  $\mathbf{G} = place(\Phi, \Gamma, p)$. This function places the desired closed-loop poles p by computing a state-feedback gain matrix **G**. Here Φ and Γ are discrete-time states and input matrices.

  NOTE: Please refer to lecture notes for the detailed algorithm and MATLAB help for the above function.
- Incorporate bounds on manipulated inputs by clipping.

LQOC CONTROLLER IMPLEMENTATION

A: Steady State Controller Gain

In this implementation, use $\mathbf{G}_\infty$ as the controller gain where $\mathbf{G}_\infty$ is the steady state solution of the Riccati equation.

- **Weighting matrices:**
  $W_x = I_{4\times4}$
  $W_u = 0.1 \times I_{2\times2}$
  where I is identity matrix of a given dimension.

- Solution for Algebraic Riccati Equation (ARE) can be found using MATLAB Control System Toolbox function *dlqr* as follows
  $[\mathbf{G}_\infty, \mathbf{S}_\infty, E_v] = dlqr(\Phi, \Gamma, W_x, W_u)$

  NOTE: Please refer to lecture notes for the detailed algorithm and MATLAB help for above function.

B: Time Varying Controller Gain Matrix
In this implementation, solve for $\mathbf{S}_k, \mathbf{G}_k$ backwards in time as discussed in class and slides. Store the $\mathbf{G}_k$ matrices. Once you have reached time instant 1 in Matlab starting from the time instant $N-1$, i.e. you have stored values of $\mathbf{G}_{N-1}, \mathbf{G}_{N-2}, ..., \mathbf{G}_1$, then implement the controller marching forward in time. The weighting matrices are the same as that specified for the case when $\mathbf{G}_\infty$ has to be used. Take $W_N = W_x$.

MPC IMPLEMENTATION

Tuning parameters for MPC

- **Prediction horizon:** $p = 50$ samples.
- **Control horizon:** $q = 20$ samples.
- **Weighting matrices:**
  $W_x = I_{4\times4}$
  $W_u = 0.1 \times I_{2\times2}$
  where I is an identity matrix of a given dimension.
- **Initial input**, $U(0) = U_s$. This initial input will be needed to implement ΔU constraints in the first window. Towards this end, use this initial input and the initial state to generate state at time instant 1. Then apply MPC starting from time instant 1.
  Note: In matlab, the starting index is 1 and not 0. So accordingly initial time instant will be 1, and the time instant for the first MPC implementation will be 2.

MPC Optimization Problem

The model predictive control problem at the sampling instant k is defined as a constrained optimization problem whereby the future manipulated input moves

$$U_{f,k} = \{u(k+j|k) : j = 0, 1, ...p-1\} \tag{5}$$

are determined by minimizing a cost function defined over prediction horizon p. Note that use of control horizon $q < p$ implies inclusion of the following constraints on the future manipulated inputs

$$u(k+j) = u(k+q-1), \quad j = q, q+1, ..., p-1 \tag{6}$$

Conversion of the MPC formulation to a quadratic programming optimization problem was discussed in detail in the class. The constraints arising from use of control horizon $q < p$ can be incorporated as additional equality constraints.

The MPC formulation as a quadratic programming problem can be written as

$$\arg\min_{U_{f,k}} J = \frac{1}{2}(U_{f,k})^T H (U_{f,k}) + (F_k)^T U_{f,k}$$

s.t.

$$\widetilde{A} U_{f,k} \leqslant \widetilde{b}$$
$$A_{eq} U_{f,k} = b_{eq}$$

where, construction of $\widetilde{A}, \widetilde{b}$ was discussed in the class. Construct A_{eq}, b_{eq} to incorporate the equality constraints arising from the use of control horizon which is less than prediction horizon.

NOTE: The MATLAB inbuilt command 'quadprog' can be used to solve the above quadratic programming problem. $x = quadprog(H, f, \widetilde{A}, \widetilde{b}, Aeq, beq)$. Please refer to MATLAB help to understand how to use the command.

Performance index calculation

- **Sum Squared Error**
  $SSE_i = \sum_{k=1}^{N_s}(X_i(k) - R)^2$, for $i = 1, 2, 3, 4$.
  Calculate the SSE value for PPC, LQOC and MPC controller implementation.

- **Sum Squared Manipulated variables**
  $SSMV_i = \sum_{k=1}^{N_s}(U_i(k) - U_s)^2$, for $i = 1, 2$.
  This is a measure of perturbations in the manipulated variables (a proxy for control effort).

Plots to generate

1. Plot states of Nonlinear plant obtained using PPC, LQOC-Steady State, LQOC-TimeVarying, and MPC controller, alongwith setpoint for each state in the same plot using 'hold on'. You can have one figure per state. Don't forget to insert a legend.

2. Plot the manipulated inputs given for the various controllers in the same plot using 'hold on'. You can have one figure per input.

MATLAB Code To Submit

- Submit all files in a zipped folder. The submitted folder should have the following MATLAB files:

  - File for PPC controller implementation.
  - File for LQOC-steady state controller implementation.
  - File for LQOC-time-varying controller implementation.
  - All files for MPC controller implementation.
  - Function file of system dynamics.
  - MAT file for continuous-time linear state space model. (Use the MAT file provided in Assignment 3).
  - File, which on execution, will generate the various plots of controlled outputs and Manipulated inputs and return the value of performance indices.

  Keep the file name as *Assn_03_ROLLNO.m* where ROLLNO is your roll number. [**Get in touch with TA for any issues.**]

- Make sure that the plots are labelled properly with axes names, titles, and legends.

Learning is fun. Best of Luck!