Model Predictive Control - EL2700

Assignment 3 : Linear Quadratic Regulator

2019

Automatic control
School of Electrical Engineering and Computer Science
Kungliga Tekniska Högskolan

1 Assignment 3: Linear-quadratic regulator design (LQR)

In this task, we will implement LQR controller for the reference tracking, then add integral action to this controller, and finally design an output feedback controller based on Kalman filter. To simplify the task, we use the linear model of the inverted cart pendulum to test the controllers. You are encouraged (but not obliged) to also perform the bonus task. This part of the assignment allows you to go back to the non-linear model and investigate if the performances are still satisfied.

PART I: LQR Implementation

We design LQR controller based on the discrete-time linearized dynamics

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + Bu_t + B_w w_t \\ y_t &= C\mathbf{x}_t \end{aligned} \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^4$, $u_t \in \mathbb{R}$, $w_t \in \mathbb{R}$, and $y_t \in \mathbb{R}$ being system state, control input, disturbance input, and output respectively. The matrices A, B, B_w, and C are provided, and the objective is to design an LQR controller with feed forward term that achieves the following three criteria:

1. It moves the cart to the reference point $r = 10$;
2. The cart should move 90% of the reference position within 10 seconds;
3. The pendulum has to be kept within $\pm 10^\circ$ around the vertical position.

Assume first $w_t = 0$. Tracking constant reference r without error is possible if there exist an equilibrium state x^{eq} and corresponding constant input u^{eq} such that

$$\mathbf{x}^{eq} = A\mathbf{x}^{eq} + Bu^{eq} \qquad\qquad r = C\mathbf{x}^{eq} \tag{2}$$

With the incremented state and control variables ($\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}^{eq}, \Delta u_t = u_t - u^{eq}$), the linear state space model of the system in (1) becomes

$$\Delta\mathbf{x}_{t+1} = A\Delta\mathbf{x}_t + B\Delta u_t \tag{3}$$

LQR controller design consists of finding the optimal gain L such that the controller $\Delta u_t = -L\Delta\mathbf{x}_t$ minimizes the quadratic cost function

$$J = \sum_{t=0}^{\infty} \Delta\mathbf{x}_t^T Q \Delta\mathbf{x}_t + \Delta u_t^T R \Delta u_t$$

where Q and R are positive (semi-) definite matrices of appropriate dimensions. To design the optimal gain L that stabilizes the system around the reference, we can use the command `dlqr`. Once L is designed, we can find u^{eq} from (2) as

$$u^{eq} = -L\mathbf{x}^{eq} + l_r r \tag{4}$$

where feed-forward gain l_r satisfies $C(I - (A - BL))^{-1}Bl_r = I$. Combining feedback from the states and feed-forward from the reference, the optimal control can be calculated as

$$u_t = -L\mathbf{x}_t + l_r r \tag{5}$$

Your tasks are as follows:

- Verify if the system is reachable and explain why is this important.
- Choose the weight matrices Q and R and design the controller (5) that meets the desired performance specification. Comment your intuition behind the final selection of Q and R. You are encouraged to try different tuning rules introduced in the lecture notes.
- Set the disturbance to zero and evaluate the performance of this controller in simulations. How does this controller track the constant reference?
- Add a small constant disturbance to mimic the presence of horizontal wind force and comment on the tracking ability of the LQR controller.

PART II: Adding integral action

Next, we add an integral action

$$i_{t+1} = i_t + h(r_t - y_t) = i_t + h(r_t - Cx_t)$$

to the controller. By incorporating the integral action to the state vector, we obtain the system

$$\begin{bmatrix} \mathbf{x}_{t+1} \\ i_{t+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -hC & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ i_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ h \end{bmatrix} r_t + \begin{bmatrix} B_w \\ 0 \end{bmatrix} w_t \tag{6}$$

To incorporate the integral state, we can modify the LQR controller as follows

$$u_t = -L\mathbf{x}_t - l_i i_t + l_r r \tag{7}$$

Since the extended system state consists of both the system state and the integral state, the gains L and l_i should be selected to minimize the cost

$$J = \sum_{t=0}^{\infty} \begin{bmatrix} \Delta\mathbf{x}_t \\ \Delta i_t \end{bmatrix}^T \begin{bmatrix} Q_1 & 0 \\ 0 & q_i \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_t \\ \Delta i_t \end{bmatrix} + \Delta u_t^T R \Delta u_t$$

where Q_1, R and q_i are positive definite weight matrices.

- Form the augmented system matrices from (6).
- Adjust the weight matrices Q_1, R and q_i and design the controller (7) that meets the desired performance specification. Motivate your choice of the weight matrices.
- Comment if the controller is eliminating the step disturbance.

PART III: Output feedback controller

We will now implement LQG controller by combining LQR controller with Kalman filter. The only formal requirement for state estimation is that the system is observable. For our cart-pendulum system, probable locations for output matrices are shown in Table 1.

- Check observability for all the three choices from Table 1 and find out possible candidates for output matrix. Use the MATLAB command `obsv` to compute observability matrix.

Number of sensors	Sensor placement	Output matrix C_p
1	cart position	$C_p = [\ 1\ 0\ 0\ 0\ ]$
1	pendulum Angle	$C_p = [\ 0\ 0\ 1\ 0\ ]$
2	both cart position and pendulum angle	$C_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Table 1: Sensor placement

We will use Kalman filter SIMULINK block to implement the estimator. The inputs to this block are: (i) State space model of the system with process noise (w_t) and measurement noise (v_t)

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + \begin{bmatrix} B & B_w \end{bmatrix} \begin{bmatrix} u_t \\ w_t \end{bmatrix} \\ y_t &= C\mathbf{x}_t + \begin{bmatrix} D & D_w \end{bmatrix} \begin{bmatrix} u_t \\ w_t \end{bmatrix} + v_t \end{aligned} \tag{8}$$

(ii) Initial guess of the state ($\mathbf{x}_0$); (iii) Process and measurement noise covariance matrices (Q_p, R_n). With the estimated state ($\hat{\mathbf{x}}_t$), we can formulate output feedback controller as:

$$u_t = -L\hat{\mathbf{x}}_t - l_i i_t + l_r r \tag{9}$$

Your tasks are as follows:

- Adjust the process and measurement noise covariance matrices (Q_p, R_n) to ensure good estimation performance of the Kalman filter. As a measure of the performance, we use the root-mean-square error (RMSE) over T time steps

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

  where $\hat{y}_t$ are the measurements predicted by an estimator, and y_t is the measured value. RMSE is plotted by running the script `run_simulation.m`.

- Compare the performance of the output feedback controller to the state feedback one. Try different feasible choices of the matrix C_p from Table 1.

- Comment on the robustness of the output feedback controller.

PART IV: Bonus Task

In this task, which is voluntary, we will evaluate the performance of the controllers (5), (7), and (9) to the non-linear cart pendulum model. Your tasks are as follows:

- How is the performance of the controller (5)? Does the controller meet the desired performance specification?

- Set the disturbance to zero and evaluate the performance of this controller in simulations. How does this controller track the constant reference?

- Is the performance of this controller similar to that of finite-time dynamic programming formulation in the last design project? Make sure that you have used same initial states in both simulations. Compare the controllers in terms of computational time. Do not forget to include the time to generate `DP_TV_L_FS_.mat` file.

- Add a small constant disturbance in the simulation and comment on the tracking ability of this controller.

- Is the controller (7) is eliminating the step disturbance? How is the performance of this controller in terms of overshoot and settling time as compared to the linear one?

- Is this controller performing better as compared to finite-time dynamic programming formulation with disturbance? Make sure that you have used same initial states in both simulations. Do not forget to include distubance in your dynamic programming simulation.

- How is the performance of the output feedback controller (9)?

- As the linear model is valid near the linearizing point, i.e. upright position of the pendulum, set the initial state $\mathbf{x}_0 = [0 \quad 0 \quad 0 \quad 0]^T$ in the script `prepare_sim.m` under the simulation folder and redo your simulation. Is the output feedback controller now performing better?

- If you fail to get a satisfatory performance, investigate what might be the problem. Is the problem related to the controller you have designed or your estimator?

- If you think that the problem is caused by your estimator, recommend some alternative estimation approaches.

PART V: Submission

To complete this design project, you should upload filled MATLAB skeleton file `assignment_3.m` to Canvas. For bonus task, you should upload filled MATLAB skeleton file `assignment_3b.m` to Canvas. Please do not upload any additional files.

Good Luck!