Optimal Control

1. Multi-input and multi-output systems, it is difficult to know all the design parameters because limited number of these can be determined from pole placement technique.
2. The matlab command “place” uses additional conditions to determine the design parameters for the system, thus designs through pole placement cannot be alone.
3. We did not know a priori about the pole locations which can give the desired performance.
4. The above ambiguities can be ignored though the use of optimal control- design parameters can be determined for multi input and output system.
5. Optimal control gives the best possible control system for a given set of performance objectives.
6. The optimal control system can be designed to meet the desired performance objectives with smallest control energy. Thus optimal control minimizes the cost associated with generating control inputs.
7. In pole placement technique, the desired performance is indirectly achieved through location of closed loop poles, whereas, the optimal control does the job directly while minimizing the control energy. Hence, an objective function needs to be designed and that to be minimized in the design process.
8. The total energy can be defined as the combination of transient and control energy.
9. By including the transient energy, we can specify the values of the acceptable maximum overshoot and settling time.

Optimal Control (LQR and LQG)

The theory of optimal control is concerned with operating a dynamic system at minimum cost. The case where the system dynamics are described by a set of linear differential equations and the cost is described by a quadratic function is called the LQ problem. One of the main results in the theory is that the solution is provided by the **linear–quadratic regulator (LQR)**, a feedback controller whose equations are given below. The LQR is an important part of the solution to the LQG (linear–quadratic–Gaussian) problem. Like the LQR problem itself, the LQG problem is one of the most fundamental problems in control theory.

LQR Controller Design

$$\dot{x}(t) = Ax(t) + Bu(t); x(t=0) = x_0$$

Square, symmetric state weighting matrix

Square, symmetric control cost matrix

We define a quadratic cost function

$$J = \int_0^\infty \left[x^T Q x + u^T R u \right]$$

By minimizing the cost function or quadratic, possible to drive the system states to the origin or equilibrium point. To achieve this, the following conditions should satisfy:

1. The entire state vector $x(t)$ is available for feedback.
2. $[A \;\; B]$ is stabilizable and $[A \;\; C]$ is detectable.
3. $R = R^T > 0$

Based on the above assumptions:

1. $u(t) = -Kx(t) \;\; with \;\; K = R^{-1}B^T P$
2. $PA + A^T P - PBR^{-1}B^T P + Q = 0$
3. $\dot{x}(t) = [A - BK]x(t)$

Which helps to minimize the cost function J

P is unique, symmetric, positive semi-definite solution to the algebric Riccati equation

LQR Controller Design

Derivation of Riccati Equation

$$\lambda(t) = P(t)x(t)$$

$$\begin{aligned}
\dot{\lambda}(t) &= \dot{P}x(t) + P\dot{x}(t) \\
&= \dot{P}x(t) + P(Ax + Bu) \\
&= \dot{P}x(t) + P(Ax - BKx(t)) \\
&= \dot{P}x(t) + P(Ax - BR^{-1}B^T Px(t)) \\
&= (\dot{P} + PA - PBR^{-1}B^T P)x
\end{aligned}$$

$$\begin{aligned}
&-(Qx + A^T\lambda) = (\dot{P} + PA - PBR^{-1}B^T P)x \\
&-Qx - A^T Px = (\dot{P} + PA - PBR^{-1}B^T P)x \\
&\Rightarrow \dot{P} + PA - PBR^{-1}B^T P + A^T P + Q = 0
\end{aligned}$$

For constant Q and R, $\dot{P} \to 0 \; for \; all \; t$ and ARE becomes

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

LQR Controller Design

How we choose LQR weights

1. Simplest choice: $Q = I, R = \rho I$. Vary ρ to get something that has good response
2. Diagonal Weights:

$$Q = \begin{bmatrix} q_1 & \cdots & \cdot\cdot \\ & & \\ & & q_n \end{bmatrix} \qquad R = \rho \begin{bmatrix} r_1 & & \\ & \cdots. & \\ & & r_n \end{bmatrix}$$

3. Output Weighting: Let $z = Hx$ be the output you want to keep small. Assume (A, H) observable. Use $Q = H^T H, R = \rho I$
4. Trial and Error

Example-Optimal Control of Double Integrator

Consider a system with cost function

$$\ddot{y} = u; \;\; J = \int_0^\infty \left[y^2(t) + ru^2(t) \right] dt$$

State-space representation of the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

The cost function can be written as

$$\ddot{y} = u; \;\; J = \int_0^\infty \left[x^T Q x + ru^2(t) \right] dt$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \;\; C^T C = Q$$

Let $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$

Now using A,B,C, and P, the ARE Becomes

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

$$-\frac{1}{r} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = 0$$

$$\Rightarrow -\frac{1}{r} p_{12}^2 + 1 = 0; \;\; p_{11} - \frac{1}{r} p_{12} p_{22} = 0;$$

$$2p_{12} - \frac{1}{r} p_{22}^2 = 0$$

Solving

$$p_{11} = \sqrt{2} r^{\frac{1}{4}}; p_{12} = \sqrt{2}; p_{22} = \sqrt{2} r^{\frac{3}{4}}$$

$$P = \begin{bmatrix} \sqrt{2} r^{\frac{1}{4}} & \sqrt{2} \\ \sqrt{2} & \sqrt{2} r^{\frac{3}{4}} \end{bmatrix}$$

LQR Controller Design- Longitudinal motion of Aircraft

Lets consider the longitudinal motion of a flexible bomber aircraft as:

$$A = \begin{bmatrix} 0.4158 & 1.025 & -0.00267 & -0.0001106 & -0.08021 & 0 \\ -5.5 & -0.8302 & -0.06549 & -0.0039 & -5.115 & 0.809 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1040 & -78.35 & -34.83 & -0.6214 & -865.6 & -631 \\ 0 & 0 & 0 & 0 & -75 & 0 \\ 0 & 0 & 0 & 0 & 0 & -100 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 75 & 0 \\ 0 & 100 \end{bmatrix}; \; c = \begin{bmatrix} -1491 & -146.43 & -40.2 & -0.9412 & -1285 & -564.66 \\ 0 & 1.0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The inputs are desired elevator deflection (u_1) and desired canard deflection (u_2), while the output are the normal acceleration (m/s^2), $y_1(t)$, and pitch rate (rad/s), y_2. Let us design an optimal regulator which would produce a maximum overshoot less than ± 2 m/s2 in the normal acceleration and less than $\pm$ 0.03 rad/s in the pitch rate, settling time less than 5s. The initial condition is given by $x(0) = [0; 0.1; 0; 0; 0]$.

LQR Controller Design- Longitudinal motion of Aircraft

The overshoots and settling time on states and controls can be determined by the appropriate values of Q and R .

Note that the existence of a unique, positive definite solution to the algebraic Riccati equation will be guaranteed if Q and R are positive semi-definite and positive definite, respectively, and the plant is controllable.

```
>>rank(ctrb(A,B)) <enter>

ans=
  6
```

Hence, the plant is controllable. By choosing $\mathbf{Q} = \mathbf{I}$, and $\mathbf{R} = \mathbf{I}$, we are ensuring that both are positive definite. Therefore, all the sufficient conditions for the existence of an optimal solution are satisfied. For solving the algebraic Riccati equation, let us use the MATLAB command *lqr* as follows:

```
>>[Ko,Mo,E]=lqr(A,B,eye(6),eye(2)) <enter>
```

To see whether this design is acceptable, we calculate the initial response of the closed-loop system as follows:

```
>>sys1=ss(A-B*Ko,zeros(6,2),C,zeros(2,2));<enter>

>>[Y1,t1,X1]=initial(sys1,[0.1 zeros(1,5)]'); u1=-Ko*X1'; <enter>
```

LQR Controller Design- Longitudinal motion of Aircraft

Let us try another design with $\mathbf{Q} = 0.01\mathbf{I}$, and $\mathbf{R} = \mathbf{I}$. As compared with the previous design, we are now specifying that it is 100 times *more important* to minimize the total control energy than minimizing the total transient energy. The new regulator gain matrix is determined by re-solving the algebraic Riccati equation with $\mathbf{Q} = 0.01\mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ as follows:

```
>>[Ko,Mo,E] = lqr(A,B,0.01*eye(6),eye(2)) <enter>
```

The closed-loop state-space model, closed-loop initial response and the required inputs are calculated as follows:

```
>>sys2=ss(A-B*Ko,zeros(6,2),C,zeros(2,2)); <enter>

>>[Y2,t2,X2] = initial(sys2,[0.1 zeros(1,5)]'); u2=-Ko*X2'; <enter>
```