Linear Quadratic Regulator

Karthik Sathyanarayanan

I. Part-A

A. Task A.1

We construct the state feedback controller by the equation:

$$P = Q + A^T PA - A^T PB(R + B^T PB)^{-}1B^T PA$$

Here A = 1, B =1, R = 1, C = 1, Q = C*C' = 1.

Plugging in the above values and simplifying it, we get :

$$(\rho + P)^{-}1 = \frac{1}{P^2} \qquad \text{eq [1]}$$

$$P^2 = \rho + P$$

$$P^2 - P - \rho = 0$$

By solving the quadratic equation we get:

$$P = \frac{1 \pm \sqrt{1+4\rho}}{2\rho}$$

For finding K , we plug in the values in the formula :

$$K = -(R + B^T PB)^{-1} B^T PA$$
$$= -(\rho + P)^{-1} P$$

Substituting the value of $(\rho + P)^{-1}$ from eq [1] , we get:

$$K = \frac{-1}{P}$$

$$K = \frac{-2}{1 \pm \sqrt{1 + 4\rho}}$$

Similarly, we know that $U_t = Kx_k$
Plugging in the values, we get:

$$U_t = \frac{-2}{1 \pm \sqrt{1 + 4\rho}} x_k$$

B. Task A.2

The transfer function can be found by solving the matrix equation of the system given :

$$\frac{(1/z - 1)}{1 + (K/z - 1)}$$

For closed loop poles , we set the denominator to 0 and simplifying further we get

$$z - 1 + K = 0$$

Substituting the value of K, we get the value from above in the equation we get :

$$Z = 1 + \frac{-1 + \sqrt{1+4\rho}}{2\rho}$$

C. Task A.3

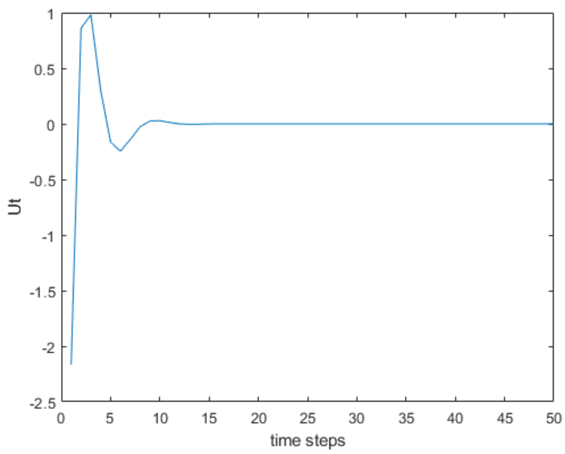
When ρ = 0 and we increase it to to 1,2,3.. and increase it gradually , the system gets unstable .
When ρ goes from infinity to 0 . the system goes from unstable to stable.
When ρ = 0 ; K= 1 and when ρ = infinity ; k= infinity.

II. Part-B

A. Task B.1

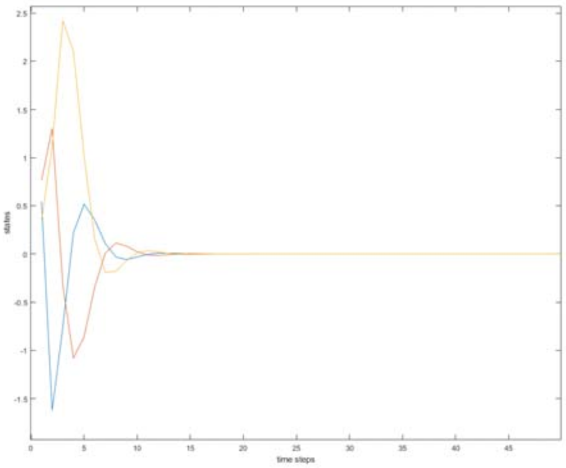
The standard finite horizon LQR design is made by solving the algebraic Riccati solution manually in a loop for 50 time steps. The initial condition is assigned and by dynamic programming approach a LQR design is created.

- Optimal input as a function of time.



The above figure illustrates the optimal input as a function of time. We can see that the optimal input oscillates until it reaches a steady state after 10 seconds.

- Resulting state as a function of time.



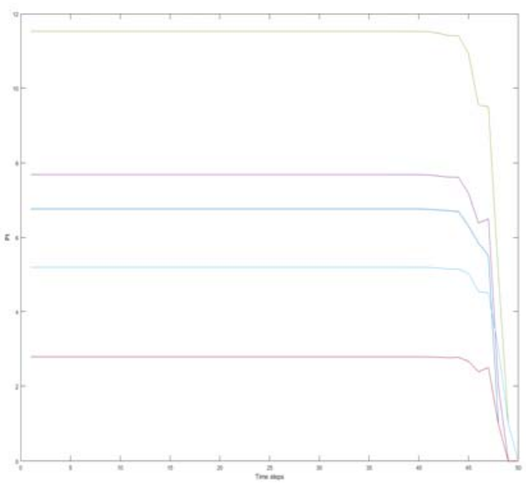
The above figure illustrates the resulting plant states as a function of time. We assign the initial state as

$$x_0 = \begin{bmatrix} 0.5428 \\ 0.7633 \\ 0.3504 \end{bmatrix}$$

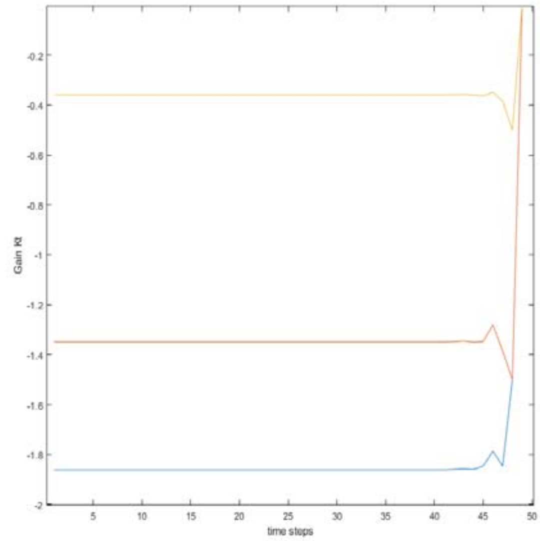
The states oscillate for 10 seconds before reaching the value of zero as the steady state when simulated for 50 time steps.

- Elements of P_t as a function of time .

The elements of P_t are traced back to find the optimal solution using the LQR design. We assign $P_N = Q_f$. and



simulate it for 50 time steps. The plot shows the 5 elements of the symmetric 3x3 matrix.



- Feedback gain K_k as a function of time .

The above diagram illustrates the gain K_k as a function of time for the plant states. We simulate K for 50 time steps by using the equation :

$$Kt = -(R + B^T PtB)^{-1} B^T PtA$$

B. Task B.2

The steady state LQR solution is calculated using dare function in MATLAB and it is used to control the system. The steady state solution of infinite horizon matches the finite horizon optimal LQR steady state solution

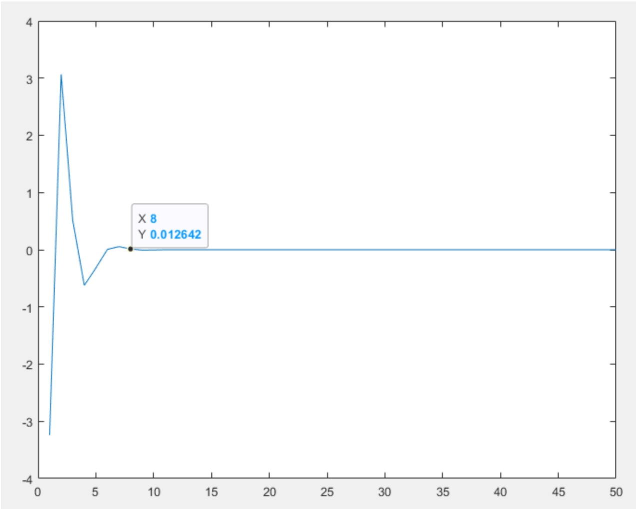
```
val(:,:,1) =

    5.7085    8.0276    3.6850
    8.0274   11.2886    5.1820
    3.6851    5.1821    2.3788
```

The figure above shows the elements of the cost function matrix for infinite horizon and the cost function for optimal LQR. We can clearly see that the finite horizon model uses same resources to reach the steady state than the in optimal LQR model.

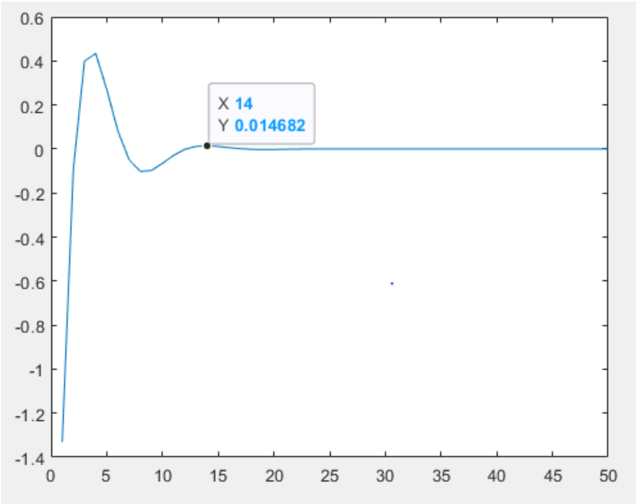
C. Task B.3

When the value of ρ is set to 0.1, the input stabilizes at 8 seconds compared to the 10 seconds.



When ρ increases the system uses less resources to reach to reach the steady state. The disadvantage of using less resources are that it reaches the steady state at longer timespans.
The converse can be said when ρ decreases. The system uses more resources to reach steady state but it reaches it at shorter timespans.

When the values of ρ is set to 10, the input stabilizes at 14 seconds.



Performance when ρ = 10

```
val(:,:,1) =

   19.3401   18.0995    6.1766
   27.1966   25.4520    8.6858
   12.4849   11.6840    3.9873
```

Performance when ρ = 0.1

```
val(:,:,1) =

    2.3126    4.5356    2.5891
    3.2521    6.3781    3.6408
    1.4929    2.9279    1.6714
```