Updated: Friday, October 31, 2008

Linear Quadratic Regulator (LQR) State Feedback Design

A system can be expressed in state variable form as

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

with $x(t) \in R^n, u(t) \in R^m$. The initial condition is $x(0)$. We assume here that all the states are measurable and seek to find a state-variable feedback (SVFB) control

$$u = -Kx + v$$

that gives desirable closed-loop properties. The closed-loop system using this control becomes

$$\dot{x} = (A - BK)x + Bv = A_c x + Bv \qquad (1)$$

with A_c the closed-loop plant matrix and v(t) the new command input.

Note that the output matrices C and D are not used in SVFB design.

If there is only one input so that m=1, then Ackermann's formula gives a SVFB K that places the poles of the closed-loop system as desired. However, it is very inconvenient to specify all the closed-loop poles, and we would also like a technique that works for any number of inputs.

The conditions under which biological life can survive are very narrow. In the course of evolution, changes of a few degrees in temperature have exterminated entire populations. Since the energy sources of living organisms, such as the cell, are quite limited, many biological organisms have evolved in such a way that they have become *optimal* in the sense of using the least required control effort to maintain their homeostasis, or equilibrium. With very little effort in pumping appropriate chemicals through the cell membrane, the cell can maintain the voltage difference across the membrane necessary for its survival and correct functioning.

Since many naturally occurring systems are optimal, it makes sense to design man-made controllers to be optimal as well. To design a SVFB that is optimal, we may define the *performance index (PI)*

$$J = \frac{1}{2}\int_0^\infty x^T Qx + u^T Ru \, dt \,. \qquad (2)$$

Substituting the SVFB control into this yields

$$J = \frac{1}{2}\int_0^\infty x^T (Q + K^T RK)x \, dt \,. \qquad (3)$$

We assume that input v(t) is equal to zero since our only concern here are the internal stability properties of the closed-loop system.

The objective in optimal design is to select the SVFB K that *minimizes* the performance index J.

The performance index J can be interpreted as an energy function, so that making it small keeps small the total energy of the closed-loop system. Note that both the state x(t) and the control input u(t) are weighted in J, so that if J is small, then neither x(t) nor u(t) can be too large. Note that if J is minimized, then it is certainly finite, and since it is an infinite integral of x(t), this implies that x(t) goes to zero as t goes to infinity. This in turn guarantees that the closed-loop system will be *stable.*

The two matrices Q (an $n \times n$ matrix) and R (an $m \times m$ matrix) are selected by the design engineer. Depending on how these design parameters are selected, the closed-loop system will exhibit a different response. Generally speaking, selecting Q large means that, to keep J small, the state x(t) must be smaller. On the other hand selecting R large means that the control input u(t) must be smaller to keep J small. This means that larger values of Q generally result in the poles of the closed-loop system matrix $A_c = (A - BK)$ being further left in the s-plane so that the state decays faster to zero. On the other hand, larger R means that less control effort is used, so that the poles are generally slower, resulting in larger values of the state x(t).

One should select Q to be *positive semi-definite* and R to be *positive definite*. This means that the scalar quantity $x^T Q x$ is always positive or zero at each time t for *all functions x(t)*, and the scalar quantity $u^T R u$ is always positive at each time t for *all* values of u(t). This guarantees that J is well-defined. In terms of eigenvalues, the eigenvalues of Q should be non-negative, while those of R should be positive. If both matrices are selected diagonal, this means that all the entries of R must be positive while those of Q should be positive, with possibly some zeros on its diagonal. Note that then R is invertible.

Since the plant is linear and the PI is quadratic, the problem of determining the SVFB K to minimize J is called the *Linear Quadratic Regulator (LQR)*. The word 'regulator' refers to the fact that the function of this feedback is to regulate the states to zero. This is in contrast to tracker problems, where the objective is to make the output follow a prescribed (usually nonzero) reference command.

To find the optimal feedback K we proceed as follows. Suppose there exists a constant matrix P such that

$$\frac{d}{dt}(x^T P x) = -x^T (Q + K^T R K) x \,. \tag{4}$$

Then, substituting into equation (3) yields

$$J = -\frac{1}{2}\int_0^\infty \frac{d}{dt}(x^T P x)\, dt = \frac{1}{2} x^T(0) P x(0) \,, \tag{5}$$

where we assumed that the closed-loop system is stable so that x(t) goes to zero as time t goes to infinity. Equation (5) means that J is now independent of K. It is a constant that depends only on the auxiliary matrix P and the initial conditions.

Now, we can find a SVFB K so that assumption (4) does indeed hold. To accomplish this, differentiate (4) and then substitute from the closed-loop state equation (1) to see that (4) is equivalent to

$$\dot{x}^T Px + x^T P\dot{x} + x^T Qx + x^T K^T RKx = 0$$

$$x^T A_c^{\ T} Px + x^T PA_c x + x^T Qx + x^T K^T RKx = 0$$

$$x^T (A_c^{\ T} P + PA_c + Q + K^T RK)x = 0$$

It has been assumed that the external control v(t) is equal to zero. Now note that the last equation has to hold for every x(t). Therefore, the term in brackets must be identically equal to zero. Thus, proceeding one sees that

$$(A - BK)^T P + P(A - BK) + Q + K^T RK = 0$$

$$A^T P + PA + Q + K^T RK - K^T B^T P - PBK = 0$$

This is a *matrix quadratic equation*. Exactly as for the scalar case, one may complete the squares. Though this procedure is a bit complicated for matrices, suppose we select

$$K = R^{-1} B^T P \ . \qquad (6)$$

Then, there results

$$A^T P + PA + Q + (R^{-1} B^T P)^T R(R^{-1} B^T P) - (R^{-1} B^T P)^T B^T P - PB(R^{-1} B^T P) = 0$$

$$A^T P + PA + Q - PBR^{-1} B^T P = 0 \ . \qquad (7)$$

This result is of extreme importance in modern control theory. Equation (7) is known as the *algebraic Riccati equation (ARE)*. It is named after Count Riccati, an Italian who lived in the 19th century and used a similar equation in the study of heat flow. It is a matrix quadratic equation that can be solved for the auxiliary matrix P given (A,B,Q,R). Then, the optimal SVFB gain is given by (6). The minimal value of the PI using this gain is given by (5), which only depends on the initial condition. This mean that the cost of using the SVFB (6) can be computed from the initial conditions *before the control is ever applied to the system*.

The design procedure for finding the LQR feedback K is:

- Select design parameter matrices Q and R
- Solve the algebraic Riccati equation for P
- Find the SVFB using $K = R^{-1} B^T P$

There are very good numerical procedures for solving the ARE. The MATLAB routine that performs this is named *lqr(A,B,Q,R)*.

The LQR design procedure is *guaranteed to produce a feedback that stabilizes the system* as long as some basic properties hold:

> **LQR Theorem.** Let the system *(A,B)* be reachable. Let R be positive definite and Q be positive definite. Then the closed loop system *(A-BK)* is asymptotically stable.

Note that this holds regardless of the stability of the open-loop system. Recall that reachability can be verified by checking that the reachability matrix $U = [B \ \ AB \ \ A^2 B \cdots A^{n-1} B]$ has full rank n.

In fact the following milder form of the LQR theorem holds. The square root of a positive semidefinite matrix Q is defined as a matrix $\sqrt{Q}$ such that $Q = \sqrt{Q}^T \sqrt{Q}$. Square roots of a positive semidefinite matrix always exist.

> **LQR Theorem 2.** Let the system *(A,B)* be stabilizable. Let R be positive definite, Q be positive semi definite, and $(A, \sqrt{Q})$ be observable. Then the closed loop *system (A-BK)* is asymptotically stable.

The theorem is interesting, since it says that the full state must be OBSERVABLE THROUGH THE COST INTEGRAND. Note that the optimal cost, if it exists, is minimal and hence bounded. Therefore the integrand goes to zero with time

$$x(t)^T Qx(t) + u^T(t)Ru(t) \to 0$$

However,

$$xT(t)^T Qx(t) + u^T(t)Ru(t) = \left(\sqrt{Q}x(t)\right)^T \left(\sqrt{Q}x(t)\right) + u^T(t)Ru(t) \equiv z^T(t)z(t) + u^T(t)Ru(t)$$

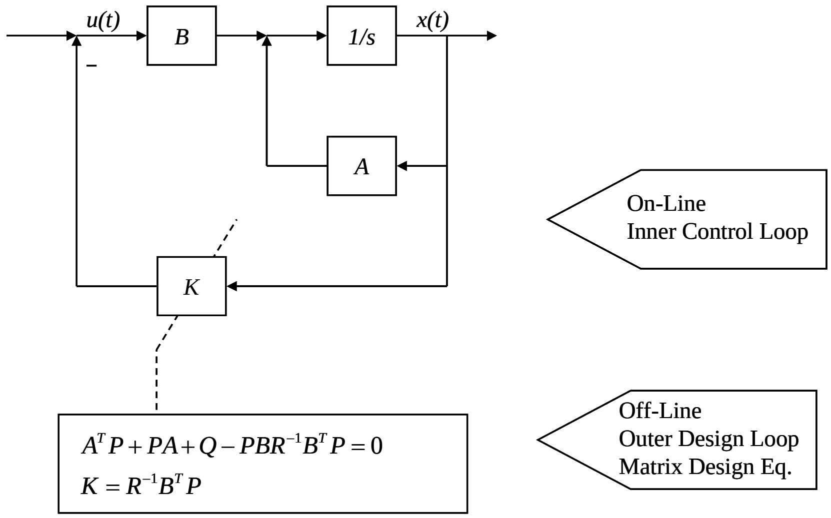
One views $z(t) = \sqrt{Q}x$ as an output of the system that is weighted in the cost function. Since R>0 this guarantees that both *u(t)* and $z(t) = \sqrt{Q}x$ go to zero. If $(A, \sqrt{Q})$ is observable, this guarantees that the full state *x(t)* goes to zero, i.e. the closed-loop system is stable.

It is important to realize that this modern control approach to feedback design is very different from the philosophy of classical control. It is characterized by

- Selecting some design matrices Q and R that are tied to the desired closed-loop performance
- Introducing an intermediate quantity P
- Solving a matrix design equation
- Obtaining a guaranteed solution that stabilizes the system
- Obtaining very little insight into the robustness or structure of the closed-loop system

In spite of the last bullet, it can be shown that the LQR has an infinite gain margin and 60 degrees of phase margin.

It is important to obtain additional robustness insight using a combination of modern control and classical design methods, as in the LQG/LTR method, which is based on singular value Bode plots.



Linear Quadratic Regulator

Example- LQR Design

The inverted pendulum is notoriously difficult to stabilize using classical techniques. Here we will use MATLAB to design a LQR for the inverted pendulum.

Open-Loop Analysis

Taking the state as $x = [p\ \dot{p}\ \theta\ \dot{\theta}]^T$, with p(t) the cart position and θ(t) the rod angle, a representative inverted pendulum is described by:

Apend =

```
0    1    0    0
0    0   -1    0
0    0    0    1
0    0    9    0
```

Bpend =

```
      0
 0.1000
      0
-0.1000
```

Cpend =

```
0    0    1    0
```

Dpend =

0

The output here is taken as the rod angle. Note that the output matrices C and D are not used in LQR SVFB design.

The open-loop poles are given by

```
» damp(Apend)

 Eigenvalue      Damping     Freq. (rad/s)

  3.00e+000    -1.00e+000     3.00e+000
  0.00e+000    -1.00e+000     0.00e+000
  0.00e+000    -1.00e+000     0.00e+000
 -3.00e+000     1.00e+000     3.00e+000
```

LQR Design #1

Select the PI weighting matrices as

```
Q =

     1     0     0     0
     0     1     0     0
     0     0    10     0
     0     0     0    10

R =

    0.1000
```

We selected Q this way so that approximately 10 times as much effort is put into keeping the angle small as keeping the cart position small. The main objective, of course, is to balance the rod.

Solving now the ARE yields the feedback gain

```
» K=lqr(Apend,Bpend,Q,R)

K =

   -3.1623  -11.1724 -235.2402  -80.1039
```

The closed-loop plant matrix is

```
» Ac= Apend-Bpend*K

Ac =

         0    1.0000         0         0
    0.3162    1.1172   22.5240    8.0104
         0         0         0    1.0000
   -0.3162   -1.1172  -14.5240   -8.0104
```

and the closed-loop poles are

```
» damp(Ac)

        Eigenvalue             Damping     Freq. (rad/s)

 -3.99e-001 + 3.46e-001i     7.56e-001      5.28e-001
 -3.99e-001 - 3.46e-001i     7.56e-001      5.28e-001
 -2.57e+000                  1.00e+000      2.57e+000
 -3.52e+000                  1.00e+000      3.52e+000
```

These are all stable with slowest time constant on the order of τ= 1/0.399=2.5 sec.

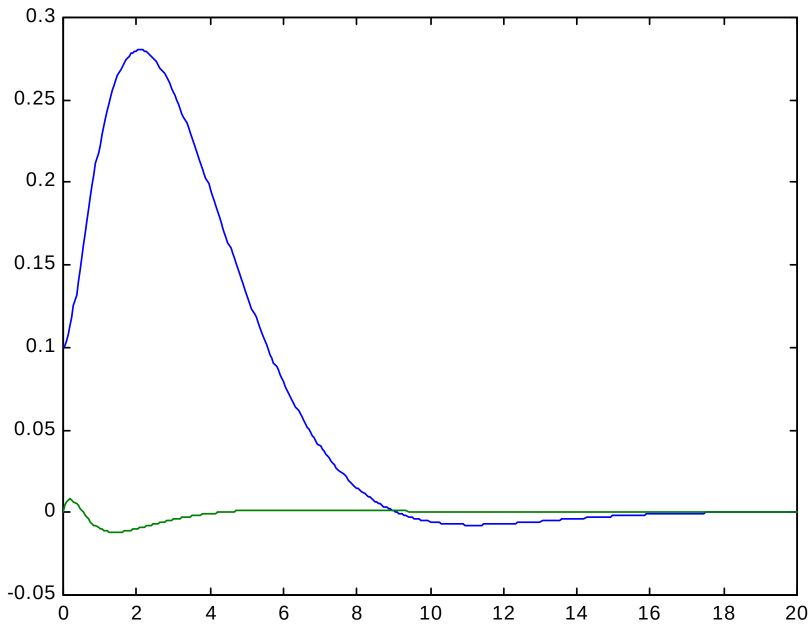
A simulation is now run using initial conditions of

```
x0=[0.1   0   0.1   0]';
```

or p(0)= 0.1 m, $\theta(0)$= 0.1 rad = 6 deg. The time period was selected as 20 sec, since the response should die out in about 5τ= 12.5 sec. The MATLAB commands are

```
t=(0:0.05:20);
» u=zeros(size(t));
» [y,x]=lsim(Ac,Bpend,Cpend,Dpend,u,t,x0);
» plot(t,y)
```

The result is shown below, where the solid line is p(t) and the dotted line is $\theta(t)$.



Note that there are two sets of poles-- two fast poles and two (complex) slow poles. From the time plot it is clear that the slow poles are associated with p(t), while the fast poles describe θ(t). This is a consequence of the fact that the angle states were weighted ten times more heavily than the position states in selecting matrix Q.

The behavior of the inverted pendulum is interesting. Note that the cart position error *increases* initially in order to catch the rod so that it does not fall over. Only after the rod is stabilized does the cart come back to the origin. (Recall that the LQ regulator is meant to keep all the states near zero.) This is bordering on intelligent behavior, and is typical of feedback control systems.

LQR Design #2

The LQR design was now redone with a different control weighting matrix of

R = 0.0100

Note that now the control weighting in the PI has been decreased by a factor of ten. This is telling the LQR that it can use more control effort than in design #1. The new SVFB gain is

```
» K1=lqr(Apend,Bpend,Q,R)

K1 =

  -10.0000  -25.4097 -308.2620 -109.4647
```

The closed-loop plant matrix is now

```
» Ac1=Apend-Bpend*K1

Ac1 =

         0    1.0000         0         0
    1.0000    2.5410   29.8262   10.9465
         0         0         0    1.0000
   -1.0000   -2.5410  -21.8262  -10.9465
```

and the closed-loop poles are

```
» damp(Ac1)

        Eigenvalue            Damping     Freq. (rad/s)

 -7.71e-001 + 5.07e-001i     8.35e-001      9.23e-001
 -7.71e-001 - 5.07e-001i     8.35e-001      9.23e-001
 -1.89e+000                  1.00e+000      1.89e+000
 -4.98e+000                  1.00e+000      4.98e+000
```

note that the poles have moved to the left when compared to design #1 so that the system is more stable. The slowest time constant is now τ= 1/0.771= 1.3 sec.

A simulation was again run and the result is shown in the figure. Note that the response is significantly faster than in the case where R= 0.1.