Feedback Control for Discrete-Time Systems

Discrete-time design for feedback controls yields Digital Controllers that can be implemented as difference equations on a digital computer.

A discrete-time system is given by

\[ x_{k+1} = Ax_k + Bu_k \]

with \( x_k \in \mathbb{R}^n \), \( u_k \in \mathbb{R}^m \). The initial condition is \( x_0 \). We seek to find a state-variable feedback (SVFB) control

\[ u_k = -Kx_k + v_k \]

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

\[ x_{k+1} = (A - BK)x_k + Bv_k = A_c x_k + Bv_k \]

with \( A_c \) the closed-loop plant matrix and \( v_k \) the new command input.

The ideas behind feedback control are the same for CT systems and DT systems, though some of the formulae change.

The equivalence of pole placement and reachability still holds, as well as the definition of stabilizability, where the requirement is only to stabilize the system using SVFB, not place all the poles. Ackermann’s formula works for both CT and DT systems.

Deadbeat Control

There is one notion that is unique to DT systems, that of deadbeat control. In deadbeat control, one selects the SVFB to place ALL POLES AT THE ORIGIN \( z=0 \). Then the closed-loop matrix \( A_c \) has the characteristic polynomial

\[ \Delta_c (z) = z^n \]

and according to the Cayley-Hamilton theorem, ones has

\[ \Delta_c (A) = A^n = 0 \]

Under this circumstance the closed-loop state trajectories are given by
\[ x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-i-1} B v_i \]

Then, if the input \( v_k = 0 \), deadbeat control yields
\[ x_n = A^n x_0 = 0. \]

That is, the state is driven to zero after \( n \) time steps.

Deadbeat control allows one to obtain behavior in sampled systems that cannot be matched using CT design. Specifically, proper selection of the sample period \( T \) and deadbeat design can allow one to achieve very fast settling times with NO OVERSHOOT. In effect, there is no tradeoff between settling time and POV using deadbeat control.

**Discrete-Time Linear Quadratic Regulator (DT LQR) State Feedback Design**

Given the discrete-time system
\[ x_{k+1} = Ax_k + Bu_k \]
we now seek to find a state-variable feedback (SVFB) control
\[ u_k = -K x_k \]
that minimizes the DT performance index
\[ J(x_k) = \frac{1}{2} \sum_{i=k}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \]
with design weighting matrices \( Q = Q^T \geq 0, R = R^T > 0 \). Note that this cost function also depends on all the future control inputs \( u_k, u_{k+1}, \ldots \) which we do not show in its argument.

This is known as the DT linear quadratic regulator problem (DT LQR), since the system is linear and the cost is quadratic.

Substituting the SVFB control into this yields
\[ J(x_k) = \frac{1}{2} \sum_{i=k}^{\infty} x_i^T (Q + K^T R K) x_i \]  \hspace{1cm} (1)

The closed-loop system using SVFB becomes
\[ x_{k+1} = (A - BK) x_k = A_k x_k \]

A difference equation equivalent to (1) is
\[ J(x_k) = \frac{1}{2} \left( x_k^T Q x_k + u_k^T R u_k \right) + \sum_{i=k+1}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \]

or
\[ J(x_k) = \frac{1}{2} \left( x_k^T Q x_k + u_k^T R u_k \right) + J(x_{k+1}) \]  \hspace{1cm} (2)
where one requires the boundary condition \( J(x_k = 0) = 0 \). That is, if one can solve (2) given a control input sequence, it is the same as finding \( J(x_k) \) for the given current state \( x_k \) by evaluating the infinite sum in (1).

Now ASSUME that the optimal cost, i.e. the minimum cost, is given for all \( k \) in the form
\[
J^*(x_k) = x_k^T P x_k
\]
That is, suppose the optimal cost is quadratic in terms of the current state and in terms of some unknown kernel matrix \( P \). If we can find the optimal feedback in terms of this assumption, then the assumption shall turn out to be valid.

Note that \( J^*(x_k) \) only depends on the initial state \( x_k \), not the future inputs, since the optimal cost is defined by selecting all future feedback controls as
\[
u_k = -K x_k
\]
with \( K \) the optimal SVFB gain.

To find the optimal SVFB and the optimal cost kernel \( P \), substitute (3) into (2) to obtain
\[
x_k^T P x_k = \frac{1}{2} \left( x_k^T Q x_k + u_k^T R u_k \right) + x_{k+1}^T P x_{k+1}
\]
Note that both \( J(x_k), J(x_{k+1}) \) are expressed in terms of the same kernel \( P \), according to the assumption (3).

Put \( x_{k+1} = A x_k + B u_k \) into this equation to obtain
\[
J(x_k) = x_k^T P x_k = \frac{1}{2} \left( x_k^T Q x_k + u_k^T R u_k \right) + (A x_k + B u_k)^T P (A x_k + B u_k)
\]
The optimal control problem is now to minimize this with respect to \( u_k \). To do this, differentiate to get
\[
0 = \frac{\partial}{\partial u_k} J(x_k) = x_k^T P x_k = R u_k + B^T P (A x_k + B u_k)
\]
This yields the optimal control as
\[
(R + B^T P B) u_k = -B^T P A x_k
\]
or
\[
u_k = -(R + B^T P B)^{-1} B^T P A x_k
\]
This defines the optimal gain as
\[
K = (R + B^T P B)^{-1} B^T P A
\]
(5)
To find \( P \), put \( u_k = -K x_k \) into (4) to get
\[
x_k^T \left[ (A - BK)^T P (A - BK) - P + Q + K^T R K \right] x_k = 0
\]
Since this must hold for all current states \( x_k \), one has the matrix equation
\[
(A - BK)^T P (A - BK) - P + Q + K^T R K = 0
\]
Note that this is a Lyapunov equation in terms of the closed-loop system matrix
\[
A_c = A - BK
\]
Now put (5) into this to obtain
\[
[A - B(R + B^T PB)^{-1} B^T PA]^T P[A - B(R + B^T PB)^{-1} B^T PA] - P + Q
+[(R + B^T PB)^{-1} B^T PA]^T R[(R + B^T PB)^{-1} B^T PA] = 0
\]

whence some simplification and determination yields
\[
A^T PA - P + Q - A^T PB(R + B^T PB)^{-1} B^T PA = 0
\]

This equation is of prime importance in modern control theory. It is known as the DT algebraic Riccati equation (DT ARE).

At this point we have succeeded in solving the DT LQR problem, therefore all our assumptions are justified.

The design procedure for finding the LQR feedback K is:

- Select design parameter weighting matrices \( Q = Q^T \geq 0, R = R^T > 0 \)
- Solve the DT algebraic Riccati equation for P
  \[
  A^T PA - P + Q - A^T PB(R + B^T PB)^{-1} B^T PA = 0
  \]
- Find the SVFB using \( K = (R + B^T PB)^{-1} B^T PA \)
- Verify performance. If not suitable try again for different weights \( Q, R \).

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

In keeping with modern design techniques, one solves a matrix quadratic equation for the auxiliary matrix P given \( (A,B,Q,R) \). Then, the optimal SVFB gain is given by (5). The minimal value of the PI using this gain is given by
\[
J^*(x_k) = x_k^T P x_k
\]
(which only depends on the current initial condition. This mean that the cost of using the optimal SVFB can be computed from the current initial conditions before the control is ever applied to the system.

Compare the DT ARE and optimal gain to the CT ARE and gain
\[
A^T P + PA + Q - PBR^{-1} B^T P = 0
\]
\[
K = R^{-1} B^T P
\]
Evidently, design equations are more complex for DT systems.

The LQR design procedure, either Ct or DT, 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 \( n \).

The following milder version of this result also holds in DT.

**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.
DISCRETIZATION OF CONTINUOUS SYSTEMS

In this section we shall discuss converting a continuous-time system into a discrete-time system. We shall explicitly include the ZOH required to convert the discrete control samples $u_k$ into the plant control input $u(t)$.

**Sampling The Plant**

In this subsection we shall show how to convert a continuous-time state-variable description of a plant into a discrete-time state-variable description.

Suppose a continuous time-invariant plant is given by

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]

with $x(t) \in \mathbb{R}^n$, measured output $y(t) \in \mathbb{R}^p$, and control input $u(t) \in \mathbb{R}^m$. For digital control purposes it is desired to define a discrete time index $k$ such that

\[ t = kT \tag{iii} \]

with $T$ the sampling period. Then, the discrete control input $u_k$ is to be switched at times $kT$, $k=0,1,...,N-1$ by the microprocessor.

The usual procedure for controlling the plant is to hold the control input $u(t)$ constant between control switchings. This may be achieved by adding a ZOH before the plant. See the figure. Then, the continuous plant input $u(t)$ is given in terms of the discrete control $u_k$ by

\[ u(t) = u_k, \ kT \leq t < (k+1)T. \tag{iv} \]

Note that $u(t)$ is switched at the times $kT$ so that it is continuous from the right.

Also shown in the figure is a sampler with period $T$ added to the output channel of the plant. This A/D device generates the samples

\[ y_k = y(kT) \tag{v} \]

of the output. Let us also define the samples
\[ x_k = x(kT) \] 

of the state vector.

It is now required to determine a dynamical relation between \( u_k \) and \( x_k \) such that

\[ x_{k+1} = A^s x_k + B^s u_k. \]  

That is, we need to determine the sampled equivalents \( A^s, B^s \) of \( A \) and \( B \).

To achieve this, first write the solution of (i), which is

\[ x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} B u(\tau) \, d\tau \]  

Setting \( t_0 = kT \) and \( t = (k+1)T \) yields

\[ x((k+1)T) = e^{AT} x(kT) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)} B u(\tau) \, d\tau \]

Since \( u(t) \) has the constant value of \( u_k \) over the sample period due to the ZOH, we may extract it from the integrand. Then, changing variables to \( \lambda = \tau - kT \) yields

\[ x_{k+1} = e^{AT} x_k + \int_{0}^{T} e^{A(T-\lambda)} B \, d\lambda \, u_k \]

Changing variables again to \( \tau = T - \lambda \) yields finally

\[ x_{k+1} = e^{AT} x_k + \int_{0}^{T} e^{A\tau} B \, d\tau \, u_k \]
By comparison with (i) we may now identify the discretized plant matrices as

\[
A^s = e^{AT}
\]

\[
B^s = \int_0^T e^{A\tau} B \, d\tau
\]

It is important to notice that the discretized plant matrix \(A^s\) is always nonsingular.

Since the output equation is a nondynamical relation, we may simply write

\[
y_k = Cx_k + Du_k.
\]

That is, the \(C\) and \(D\) matrices are unchanged on discretization.

The design of digital controls proceeds as follows. First, a continuous-time state-variable model of the plant is derived. Then, \(A^s\) and \(B^s\) are determined using the above equations. Finally, the techniques to be presented are used to design a discrete control sequence \(u_k\). During the implementation phase when the control is actually applied to the plant, \(u(t)\) is manufactured by passing \(u_k\) through a ZOH. Commercial digital signal processors (DSPs) will usually have a ZOH built in.

It is important to clearly realize that this approach to digital controls design explicitly includes the effects of the ZOH, since \(u(t)\) was assumed constant over the sample period in deriving \(B^s\). Thus, the resulting controller takes into account the properties of the hold and sampling processes, guaranteeing exact behavior at the sample instants.

**DISCRETIZATION OF PERFORMANCE INDEX**

The LQR performance index for the continuous time system (7.1.1) is

\[
J = \frac{1}{2} \int_0^\infty (x^TQx + u^TRu) \, dt
\]

The corresponding performance index for the discrete-time system (7.1.7) is

\[
J^s = \frac{1}{2} \sum_{i=k}^{n} (x_i^TQ^sx_i + u_i^TR^su_i)
\]

with

\[
Q^s = QT, \quad R^s = R/T.
\]

Note that the weighting matrices \(Q, R\) must also be discretized according to these formulas.
Example- Discrete-Time LQR Design

The inverted pendulum is notoriously difficult to stabilize using classical techniques. Here we will use MATLAB to design a discrete-time 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 $\theta(t)$ the rod angle, a representative inverted pendulum is described by the continuous-time dynamics

$$A_{\text{pend}} =
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 9 & 0
\end{bmatrix}
$$

$$B_{\text{pend}} =
\begin{bmatrix}
0 \\
0.1000 \\
0 \\
-0.1000
\end{bmatrix}
$$

$$C_{\text{pend}} =
\begin{bmatrix}
0 & 0 & 1 & 0
\end{bmatrix}
$$

$$D_{\text{pend}} =
\begin{bmatrix}
0
\end{bmatrix}
$$
We select a sampling period of $T=0.1$ sec. This is smaller than the system time constant of $1/3$ sec. Using ZOH discretization with MATLAB routine c2d, the sampled data dynamics are given by

\[
\begin{align*}
\gg [ad,bd] &= \text{c2d}(A_{\text{pends}}, B_{\text{pends}}, 0.1) \\
ad &= \\
1.0000 & 0.1000 & -0.0050 & -0.0002 \\
0 & 1.0000 & -0.1015 & -0.0050 \\
0 & 0 & 1.0453 & 0.1015 \\
0 & 0 & 0.9136 & 1.0453 \\
\end{align*}
\]

\[
bd = \\
0.0005 \\
0.0100 \\
-0.0005 \\
-0.0102 \\
\]

**LQR Design #1**

Select the continuous-time PI weighting matrices as

\[
Q = \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 10 & 0 \\
0 & 0 & 0 & 10 \\
\end{bmatrix}
\]

\[
R = \\
0.1000 \\
\]

Then, the sampled LQR weighting matrices are

\[
Q^s = QT, \quad R^s = R / T .
\]

with $T=0.1$ sec.

Now, the DT ARE is solve using MATLAB routine using 2.10
Now MATLAB routine dlsim(.) can be used to simulate the closed-loop discrete-time system. The resulting response is indistinguishable from the continuous-time LQR result shown before. The result is shown below, where the solid line is $p_k$ and the dotted line is $\theta_k$. 

![Graph showing the simulation result](image-url)