Control Synopsis for EECS 192, Spring 1999

R.S. Fearing
Dept. of EECS
UC Berkeley

1 Overview

The objective for the autonomous race car is to track a path as quickly and accurately as possible. In one view, the car has basically two *control inputs*, the motor torque and the steering position. The car can be considered to have six *states* specified by a vector $\mathbf{x} = [x\ y\ \theta\ \dot{x}\ \dot{y}\ \dot{\theta}]$, where $[xy\theta]$ specifies relative position to a local segment of the track, and $[\dot{x}\ \dot{y}\ \dot{\theta}]$ specifies relative velocity. For the car, we can consider that the state and the output $\mathbf{y}$ are the same. We can also make the problem simpler by separating the kinematic steering control from the motor velocity control.

The basic idea for a control system is shown in Figure 1, where H represents the physics (kinematics and dynamics) of the car. The designer's task is to find a good algorithm for u which gives the "best" tracking of the track. In general, the output of the system y is the solution of some differential or difference equation, such as

$$y(t) = f(x_o, u(t), t) \tag{1}$$

where x_o is the initial condition. In principle, the optimal control could be found as

$$u_{opt}(t) = g(x_o, \mathbf{x}(t),\ y_{opt}(t), t) \tag{2}$$

if one had complete information about the system and the path to be tracked. In practice, one hardly ever has this information (planetary orbital mechanics is one notable exception). Instead, one applies feedback or uses some form of look-up table approximation.

2 Feedback Control: Continuous Time

The basic principal of linear feedback control is shown in Figure 2. The *controller* tries to minimize the error signal by applying a *control input* $u(t)$ to the *plant* H. The error signal $e(t)$ is the difference between the reference $r(t)$ (otherwise known as the command input) and the sensed signal $y_s(t)$.

We can show the general function of feedback with a trivial example plant and control law. Consider a first order linear time invariant system with state x described by:

$$\dot{x}(t) = ax(t) + bu(t) \tag{3}$$

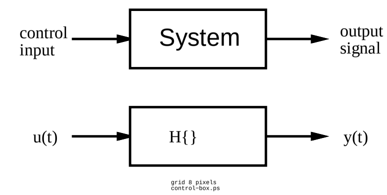

Figure 1: The basic idea behind control. Given a model of the plant H, specify the control effort $u(t)$ to get a desired behavior (or output) $y(t)$.

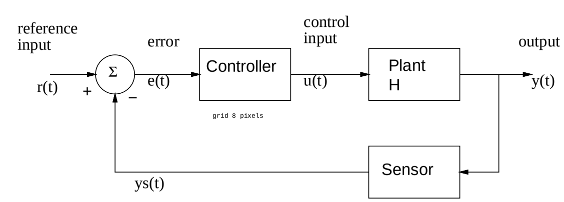

Figure 2: Basic feedback control system. Typically, the plant is modelled by differential or difference equations. Control system may be continuous or discrete time.

and output

$$y(t) = cx(t) + du(t) \tag{4}$$

with $u(t)$ the input, a, b constants and for simplicity $c = 1$ and $d = 0$. We can use a simple linear controller;

$$u(t) = k_p e(t) = k_p(r(t) \; - \; y(t)). \tag{5}$$

Then

$$\dot{x} = ax + bk_p e(t) = ax + bk_p(r - x) = (a - bk_p)x + bk_p r. \tag{6}$$

By inspection, the homogeneous response of the system (input $r(t) = 0$) is now

$$x(t) = x(0)e^{(a-bk_p)t} \quad \text{for} \quad t \geq 0. \tag{7}$$

The beauty of this controller is that, in principle, we can make the error $e(t)$ converge exponentially to zero as fast as we like.

The total solution can be found by using an integrating factor. Let $b' = bk_p$ and $a' = a - bk_p$, rearranging and premultiplying eqn. 6 by $e^{-a't}$ one gets

$$e^{-a't}[\dot{x}(t) - a'x(t)] = \frac{d}{dt}[e^{-a't}x(t)] = e^{-a't}b'u(t) \; . \tag{8}$$

Integrating both sides from 0 to t_o

$$e^{-a't_o}x(t_o) - x(0) = \int_0^{t_o} e^{-a'\tau}b'u(\tau)d\tau \tag{9}$$

or rearranging and premultiplying by $e^{a't_o}$:

$$x(t_o) = e^{a't_o}x(0) + \int_0^{t_o} e^{a'(t_o-\tau)}b'u(\tau)d\tau \; . \tag{10}$$

Note that integral in equation 10 is just convolution from EECS120. One of the most basic measures of system performance is the *step response*. So letting $r(t)$ be the unit step and $x(0) = 0$, then

$$x(t_o) = b'\int_0^{t_o} e^{a't_o}e^{-a'\tau}d\tau \;=\; \frac{-b'e^{a't_o}}{a}e^{-a'\tau}|_0^{t_o} \;=\; \frac{-b'}{a'}(1 - e^{-a't_o}) \; . \tag{11}$$

As expected, we get the classic first order response, which is stable for $a' < 0$.

2.1 Laplace Transform Viewpoint

Taking the Lapace transform of both sides of eqn. 6 one obtains:

$$sX(s) - x(0^-) = (a - bk)X(s) + bkR(s) \tag{12}$$

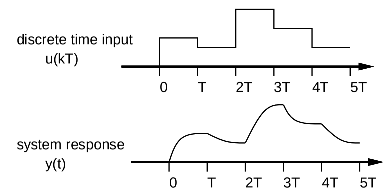

Figure 3: A discrete time control input with zero order hold applied to a continuous time system.

where $X(s), R(s)$ are the Laplace transforms of $x(t)$ and $r(t)$ respectively. Rearranging,

$$X(s) = \frac{bk}{s - a + bk}R(s) + \frac{x(0^-)}{s - a + bk} \; . \tag{13}$$

This system has a "pole" at $s = a - bk$. The system is stable if $a - bk < 0$. Note that equations 10 and 13 are equivalent.

3 Feedback Control: Discrete Time

Since we are using digital control, i.e. we change the PWM output duty cycle only once per iteration of the main real-time loop, we need to consider the effects of discrete time control. The interesting point is that the stability, even for a simple first order system, is markedly effected by the choice of sampling rate and feed back gains. Our control input model is shown in Figure 3, where the control signal $u(t)$ is now constant over the sample period T. (This corresponds to a *zero order hold* model.) We can predict the response to this input by looking at it as a superposition of step responses of the form eqn. 11. From eqn. 10, the response at time kT is just

$$x(kT) = e^{akT}x(0) + e^{akT}\int_0^{kT} e^{-a\tau}bu(\tau)d\tau \; . \tag{14}$$

It is important to note that 14 gives the *exact* solution for $x(t = kT)$. At time T later the response is just:

$$x((k+1)T) = e^{a(k+1)T}x(0) + e^{a(k+1)T}\int_0^{(k+1)T} e^{-a\tau}bu(\tau)d\tau \; . \tag{15}$$

We can get a recursive relation for $x((k+1)T)$ by premultiplying eqn. 14 by e^{aT} and subtracting from eqn. 15, obtaining:

$$x((k+1)T) = e^{aT}x(kT) + e^{a(k+1)T}\int_{kT}^{(k+1)T} e^{-a\tau}bu(\tau)d\tau \; = e^{aT}x(kT) + \int_0^{T} e^{a\lambda}bu(kT)d\lambda \; , \tag{16}$$

where for the last expression, we substituted $\lambda = (k+1)T - \tau$.

Define two constants G and H which are a function only of the sample period T, where:

$$G(T) \equiv e^{aT} \quad \text{and} \quad H(T) \equiv b\int_0^T e^{a\lambda} d\lambda \,. \tag{17}$$

Now we can rewrite our continuous time system eqn. 3 in discrete time as:

$$x((k+1)T) = G(T)x(kT) + H(T)u(kT) \tag{18}$$

and

$$y(kT) = Cx(kT) + Du(kT) \,. \tag{19}$$

It is worthwhile to convince yourself that this is the exact solution for $x(t)$ when $t = kT$.

3.1 Discrete Time Solution

We can find an equivalent formula to eqn. 10 by explicitly solving the state equations as follows:

$$x(1) = Gx(0) + Hu(0) \tag{20}$$

$$x(2) = Gx(1) + Hu(1) = G^2x(0) + GHu(0) + Hu(1) \tag{21}$$

$$x(3) = Gx(2) + Hu(2) = G^3x(0) + G^2Hu(0) + GHu(1) + Hu(2) \tag{22}$$

Thus we can write the discrete time convolution as

$$x(k) = G^k x(0) + \sum_{j=0}^{k-1} G^{k-j-1} Hu(j) \,. \tag{23}$$

For stability, the eigenvalues of G must have magnitude < 1. So if the continuous time plant is stable ($a < 0$) the discrete time plant will have eigenvalues with magnitude less than 1. As we see in the next subsection, a proportional feedback $u(kT) = k_p(r(kT) - y(kT))$ is not necessarily stable.

3.2 Discrete Time Controller

Let's see what happens to the overall system when a discrete time proportional control $u(kT) = k_p(r(kT) - x(kT))$ is added. Remember, our plant is still described by a linear differential equation (3) but now we are sensing the plant output at discrete time intervals, and changing the output at discrete intervals. We are neglecting any delay in computing the control law, which we might need to fix if our sample rate

is low compared to system response frequencies. Thus our new discrete time state equations (18) become:

$$x((k+1)T) = G(T)x(kT) + H(T)k_p(r(kT) - x(kT)) = [G - Hk_p]x(kT) + Hk_pr(kT) \,. \tag{24}$$

Recall that in the first order continuous time case, the homogeneous solution could be made stable by making the proportional feedback k_p sufficiently positive and large. In discrete time, we need to be more careful. Let's see why. Using eqn. 17 we get $H(T) = b\int_0^T e^{a\lambda}d\lambda = \frac{b}{a}(e^{aT} - 1)$, then

$$x((k+1)T) = [e^{aT} + \frac{k_p}{a}(1 - e^{aT})]x(kT) + Hk_pr(kT) \;=\; G'x(kT) + Hk_pr(kT) \;. \tag{25}$$

When is this controller stable? By inspection, the constant multiplying the $x(kT)$ term must have magnitude less than 1, or

$$|e^{aT} - \frac{k_p}{a}(e^{aT} - 1)| < 1. \tag{26}$$

If the original continuous time system is stable, then $a < 0$. Then if we look at $k_p = 0$ the system is obviously stable, since $e^{aT} < 1$. What happens as we try to increase k_p? We can increase k_p until $G' = 0$, then the response will be oscillatory for $-1 < G' < 0$, and unstable for $G' < -1$. Think about $G' = 0$ for a minute. If $G - Hk_p = 0$, this implies that you can control $x(kT)$ in one time step with no delay, the dynamics of the plant have disappeared.

3.3 Z Transform Viewpoint

Ignoring initial conditions, and assuming the existence of the *z transform* of $x(kT)$ and $u(kT)$ we have

$$zX(z) = GX(z) + HU(z) \tag{27}$$

and

$$X(z) = [zI - G]^{-1}HU(z) \quad . \tag{28}$$

4 Important Issues

The linear control paradigm is quite powerful, but it neglects the following very important issues:

- Actuator Saturation.
  With large error signals $e(t)$, the actuator will not be able to supply the required effort, for example, the motor torque is limited by battery voltage and series resistance.

- Quantization and Sensor Noise.
  With high gains, the system will respond to quantization and noise, giving erratic outputs.
- Computation Delay.
  Computation delay in discrete time systems introduces extra phase shift, decreasing stability.
- Nonlinearities.
  The linearized model may be appropriate for low speeds, but can neglect destabilizing higher order effects.
- Non-optimal.
  Linear control assumes that a planar control surface is adequate over the whole state space; control is typically far from optimal.

5 Other Control Methods

There are many approximation techniques to finding a control surface $u = g(r, y, t)$ using neural nets, fuzzy logic, or polynomials. The general function can be described as:

$$u = \text{lookup}[y][r] \tag{29}$$

where y and r are appropriately scaled and quantized versions of the system output and reference command respectively. It is interesting to note that this technique is inherently more powerful than the linear control approach. For example, all cases where $r \approx y$ do not generate the same control effort u. This could be useful for certain types of steering errors.

The big drawback to this approach is that stability is hard to verify. Most approaches to finding the parameters are quite ad hoc, and may be time consuming — now there are many more interacting parameters to tune. Fuzzy control does provide some intutitive knobs to tweak.

more to be added....

References

[1] E. Cox, "Fuzzy Fundamentals", *IEEE Spectrum,* pp. 58-61, 1992.

[2] G. Franklin et al, *Feedback Control of Dynamic Systems* , 2nd edition

[3] B. Kuo, *Automatic Control Systems*, 6th edition

[4] N. S. Nise *Control Systems Engineering*, Benjamin/Cummings, 1995.

[5] K. Ogata, *Modern Control Engineering*, 2nd edition