3.5 Discretization of Analog Controllers

So far, we have mainly discussed the step-invariant discretization, which, given a sampler and hold device, produces from a continuous-time plant G and discrete-time system $G_d = SGH$. For linear systems, we have discussed the effect of this transformation in the frequency domain by establishing the relationship between the transfer functions of the G and G_d, including the potential creation of undesired frequency components by aliasing and the introduction of a perturbation term due to the hold device. We have also derived exact state-space recurrence equations for G_d.

The discrete-time G_d captures the behavior of G at the sampling times. One of the most common ways of designing digital controllers is then to perform the design in discrete-time directly using G_d, provided we somehow verify that nothing bad happens for the resulting closed-loop system between the sampling times. For example, we can design a discrete-time controller K_d that optimizes a discrete-time performance criterion for the discrete-time closed-loop system composed of G_d and K_d, see Chapter 6. Note that optimizing a discrete-time performance criterion does not guarantee that the continuous-time behavior is satisfying in general however, see Example 6.2.1.

Continuing with our discussion of system discretization however, there are many situations where we have an analog controller design available, and we simply would like to obtain a digital implementation of it. This situation can result for example from the following facts

- for various reasons we prefer designing controller in continuous-time: e.g. because frequency domain reasoning is easier than for DT systems (PID controller design is usually discussed only in continuous-time), we do not have to worry about the choice of the sampling period at this stage (changing the sampling period changes the transfer function of the DT system) or about neglecting intersample behavior, and some calculations can be easier (.e.g. the CT Riccati equations are easier to handle).

- an analog controller was inherited from a previous system implementation, and we do not wish to redesign it (e.g. due to the cost of retesting and recertifying).

Assume therefore that a CT controller K is available, and we would like to derive from it a DT controller K_d, operating with the sampling period h. One way of doing this is to use again the step-invariant transformation and let $K_d = SKH$. Note however that here the operators S and H *do not correspond to actual physical devices*, they are just a mathematical representation of the discretization process. There is still however the physical barrier to the synchronicity assumption of these operators, coming from the computation time requirements of the processor. We continue to neglect this issue for now, but in general the mathematical H device at the input of K_d operates at the same instants as the physical S device it is connected to, and similarly at the output of K. Starting then from a state-space representation of K, we obtain then an

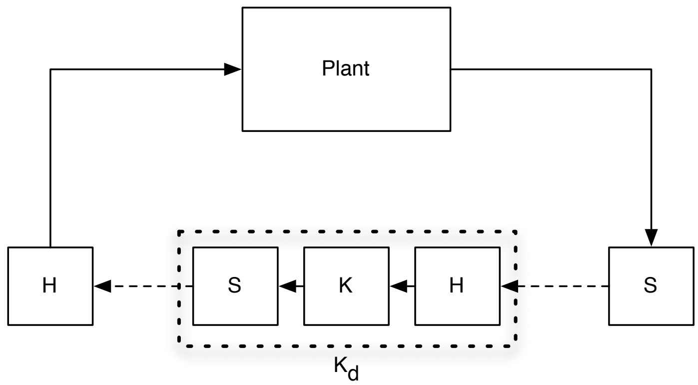

Figure 3.6: Step-Invariant Transformation for the Discretization of a Continuous-Time Controller Design. The blocks S and H that are directly connected operate at the same sampling times. The blocks S and H external to K_d correspond to physical devices, where the blocks S and H within K_d are mathematical operations producing $K_d = SKH$.

implementation of K_d in terms of difference equations as discussed in Section 3.4 (assuming K is linear).

For the plant, the discretization is dictated by the choice of sampling and hold device, and the use of the step-invariant transformation is results essentially as a consequence of these technological choices. On the controller side however, there is no such restriction and the step-invariant transformation is only one possible way of obtaining K_d from the continuous-time system K.

Bilinear Transformation

Another particularly common way of discretizing an analog controller is the bilinear transformation (also called Tustin's method). It is based on the trapezoidal approximation of integrals. Namely consider an integrator, i.e. a block with transfer function $1/s$, with input u and output y, over a sampling period

$$y((k+1)h) = y(kh) + \int_{kh}^{(k+1)h} u(\tau) d\tau.$$

We approximate this formula by

$$y((k+1)h) = y(kh) + \frac{h}{2}[u((k+1)h) + u(kh)].$$

The transfer function of this recurrence is

$$\begin{aligned}
\lambda^{-1}Y(\lambda) &= Y(\lambda) + \frac{h}{2}[\lambda^{-1}U(\lambda) + U(\lambda)] \\
\frac{Y(\lambda)}{U(\lambda)} &= \frac{h}{2}\frac{1+\lambda}{1-\lambda}.
\end{aligned}$$

This motivates the bilinear transformation

$$\begin{aligned}
\frac{1}{s} &\longleftrightarrow \frac{h}{2}\frac{1+\lambda}{1-\lambda} \\
s &\longleftrightarrow \frac{2}{h}\frac{1-\lambda}{1+\lambda}.
\end{aligned}$$

Using this change of variable, we can map a continuous-time transfer matrix $G(s)$ to a discrete-time transfer matrix $G_{bt}(\lambda)$, i.e.

$$G_{bt}(\lambda) = G\left(\frac{2}{h}\frac{1-\lambda}{1+\lambda}\right).$$

In terms of state-space models, we can derive that starting with $G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, we get $G_{bt}(\lambda) = \left[\begin{array}{c|c} A_{bt} & B_{bt} \\ \hline C_{bt} & D_{bt} \end{array}\right]$, with

$$\begin{aligned}
A_{bt} &= \left(I - \frac{h}{2}A\right)^{-1}\left(I + \frac{h}{2}A\right) \\
B_{bt} &= \frac{h}{2}\left(I - \frac{h}{2}A\right)^{-1} B \\
C_{ct} &= C(I + A_{bt}) \\
D_{dt} &= D + CB_{bt},
\end{aligned}$$

provided $2/h$ is not an eigenvalue of A. The mapping from s to λ is

$$\lambda = \frac{1 - hs/2}{1 + hs/2},$$

which maps the right half-plane into the unit disk.

Bilinear Transformation with Prewarping

Non-causal Reconstruction via Shannon's Theorem

Classical Software Implementation of a Digital Controller