Discrete-Time Systems and Discretization

Recall that in a discrete-time system, the state vector \( \mathbf{x}(t) \) evolves according to a difference equation rather than a differential equation:

\[
\mathbf{x}(t + 1) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad t = 0, 1, 2, \ldots
\]  

Here \( f(\mathbf{x}, \mathbf{u}) \) is a function that gives the state vector at the next time instant based on the present values of the states and inputs.

As in the continuous-time case, when \( f(\mathbf{x}, \mathbf{u}) \) is linear in \( \mathbf{x} \in \mathbb{R}^n \) and \( \mathbf{u} \in \mathbb{R}^m \), we can rewrite it in the form

\[
f(\mathbf{x}, \mathbf{u}) = A\mathbf{x} + B\mathbf{u}
\]

where \( A \) is \( n \times n \) and \( B \) is \( n \times m \). The state model is then

\[
\mathbf{x}(t + 1) = A\mathbf{x}(t) + B\mathbf{u}(t).
\]  

When the input \( \mathbf{u}(t) \) in (1) is a constant vector \( \mathbf{u}^* \), the equilibrium points are obtained by solving for \( \mathbf{x} \) in the equation:

\[
\mathbf{x} = f(\mathbf{x}, \mathbf{u}^*).
\]  

If \( \mathbf{x}^* \) satisfies this equation and we start with the initial condition \( \mathbf{x}^* \), the next state is \( f(\mathbf{x}^*, \mathbf{u}^*) \), which is again \( \mathbf{x}^* \). The same argument applies to subsequent time instants, so \( \mathbf{x}(t) \) remains at \( \mathbf{x}^* \).

For the linear system (2) the equilibrium condition (3) becomes:

\[
\mathbf{x} = A\mathbf{x} + B\mathbf{u}^*, \quad \text{or, equivalently} \quad (I - A)\mathbf{x} = B\mathbf{u}^*.
\]

Linearization for nonlinear discrete-time systems is performed similarly to continuous-time. The perturbation variables \( \mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}^* \) and \( \mathbf{u}(t) := \mathbf{u}(t) - \mathbf{u}^* \) satisfy:

\[
\mathbf{x}(t + 1) = \mathbf{x}(t + 1) - \mathbf{x}^* = f(\mathbf{x}(t), \mathbf{u}(t)) - \mathbf{x}^*
\]

\[
\approx f(\mathbf{x}^*, \mathbf{u}^*) + \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{u}^*) |_{\mathbf{x} = \mathbf{x}^*, \mathbf{u} = \mathbf{u}^*} \mathbf{x}(t) + \nabla_{\mathbf{u}} f(\mathbf{x}^*, \mathbf{u}^*) |_{\mathbf{x} = \mathbf{x}^*, \mathbf{u} = \mathbf{u}^*} \mathbf{u}(t) - \mathbf{x}^*.
\]

Substituting \( f(\mathbf{x}^*, \mathbf{u}^*) - \mathbf{x}^* = 0 \), which follows because \( \mathbf{x}^* \) is an equilibrium, we get

\[
\mathbf{x}(t + 1) \approx A\mathbf{x}(t) + B\mathbf{u}(t)
\]

where \( A = \nabla_{\mathbf{x}} f(\mathbf{x}^*, \mathbf{u}^*) |_{\mathbf{x} = \mathbf{x}^*, \mathbf{u} = \mathbf{u}^*} \) and \( B = \nabla_{\mathbf{u}} f(\mathbf{x}^*, \mathbf{u}^*) |_{\mathbf{x} = \mathbf{x}^*, \mathbf{u} = \mathbf{u}^*} \).
Changing State Variables

Given the state vector \( \vec{x} \in \mathbb{R}^n \) any transformation of the form

\[
\vec{z} := T \vec{x},
\]

where \( T \) is an \( n \times n \) invertible matrix, defines new variables \( z_i, i = 1, \ldots, n \), as a linear combination of the original variables \( x_1, \ldots, x_n \).

To see how this change of variables affects the state equation

\[
\vec{x}(t + 1) = A \vec{x}(t) + B \vec{u}(t),
\]

note that

\[
\vec{z}(t + 1) = T \vec{x}(t + 1) = TA \vec{x}(t) + TB \vec{u}(t)
\]

and substitute \( \vec{x} = T^{-1} \vec{z} \) in the right hand side to obtain:

\[
\vec{z}(t + 1) = TAT^{-1} \vec{z}(t) + TB \vec{u}(t).
\]

Thus the original \( A \) and \( B \) matrices are replaced with:

\[
A_{\text{new}} = TAT^{-1}, \quad B_{\text{new}} = TB. \tag{5}
\]

The same change of variables brings the continuous-time system

\[
\frac{d}{dt} \vec{x}(t) = A \vec{x}(t) + B \vec{u}(t)
\]

to the form

\[
\frac{d}{dt} \vec{z}(t) = A_{\text{new}} \vec{z}(t) + B_{\text{new}} \vec{u}(t)
\]

as depicted below.

We use particular choices of \( T \) to obtain special forms of \( A_{\text{new}} \) and \( B_{\text{new}} \) that make the analysis easier. For example, we saw in Lecture 3A that we can make \( A_{\text{new}} \) diagonal if the \( n \times n \) matrix \( A \) has \( n \) independent eigenvectors \( \vec{v}_1, \ldots, \vec{v}_n \). This is because the matrix \( V = [\vec{v}_1 \cdots \vec{v}_n] \) satisfies

\[
AV = [A \vec{v}_1 \cdots A \vec{v}_n] = [\lambda_1 \vec{v}_1 \cdots \lambda_n \vec{v}_n] = V \begin{bmatrix} \lambda_1 & \cdots & \lambda_n \end{bmatrix},
\]

\[
=: \Lambda
\]
therefore $V^{-1}AV = \Lambda$. This means that the choice

$$T = V^{-1}$$

gives $A_{\text{new}} = TAT^{-1} = \Lambda$, which is diagonal.

**Digital Control**

In upcoming lectures we will be designing the input signal $\vec{u}$ of a continuous-time system

$$\frac{d}{dt} \vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) \quad (6)$$

to ensure that the solution $\vec{x}(t)$ meets requirements, such as reaching a target state in a given amount of time.

The input signal is typically generated digitally in a computer, by using measurements of $\vec{x}(t)$ sampled every $T$ units of time. Thus the computer receives a discrete sequence

$$\vec{x}(0), \vec{x}(T), \vec{x}(2T), \cdots$$

as shown in the figure below. We use the notation

$$\vec{x}_d(k) := \vec{x}(kT) \quad (7)$$

where the subscript 'd' stands for 'discrete', so that we can represent the samples $\vec{x}(0), \vec{x}(T), \vec{x}(2T), \cdots$ as a discrete-time signal

$$\vec{x}_d(0), \vec{x}_d(1), \vec{x}_d(2), \cdots$$

Using this sequence an appropriate control algorithm generates inputs to the system, again as a discrete sequence

$$\vec{u}_d(0), \vec{u}_d(1), \vec{u}_d(2), \cdots$$

However, since the system (6) admits only continuous-time inputs, this sequence must be converted to continuous-time. This is typically done with a zero-order hold device that keeps $\vec{u}(t)$ constant at $\vec{u}_d(0)$ in the interval $t \in [0, T)$, at $\vec{u}_d(1)$ for $t \in [T, 2T)$, and so on. Therefore,

$$\vec{u}(t) = \vec{u}_d(k) \quad t \in [kT, (k+1)T), \quad (8)$$
which has a staircase shape as shown below.

The overall control scheme is illustrated below where the D/C (discrete-to-control) block represents zero-order hold and the C/D (continuous-to-discrete) block represents sampling.

Discretization

From the viewpoint of the controller, the system combined with D/C and C/D blocks (dashed box in the figure above) receives a discrete input sequence $\vec{u}_d(k)$ and generates a discrete state sequence $\vec{x}_d(k)$ that consists of snapshots of $\vec{x}(t)$.

We now wish to derive a discrete-time model

$$\vec{x}_d(k+1) = A_d \vec{x}_d(k) + B_d \vec{u}_d(k)$$

(9)

that describes how the state evolves from one snapshot to the next. That is, we want (9) to return the next sample of the continuous-time system (6) when the input $\vec{u}(t)$ is constant in between the samples.

To see how such a discrete-time model can be derived, first assume the continuous-time system has a single state and single input:

$$\frac{d}{dt}x(t) = \lambda x(t) + bu(t).$$

(10)

Since the value of $x(t)$ at $t = kT$ is $x_d(k)$, the solution of the scalar differential equation above with initial time $kT$ is

$$x(t) = e^{\lambda(t-kT)}x_d(k) + \int_{kT}^{t}e^{\lambda(t-\tau)}bu(\tau)d\tau.$$
We also know that the input \( u(t) \) from \( t = kT \) to \( t = kT + T \) is the constant \( u_d(k) \). Thus, the solution at time \( t = kT + T \) is

\[
x(kT + T) = e^{\lambda T} x_d(k) + \int_{kT}^{kT+T} e^{\lambda (kT+T-\tau)} bu_d(k) d\tau.
\]

Substituting \( x(kT + T) = x_d(k + 1) \) and factoring \( bu_d(k) \) out of the integral (since it is constant) we get

\[
x_d(k + 1) = e^{\lambda T} x_d(k) + \left( \int_{kT}^{kT+T} e^{\lambda (kT+T-\tau)} d\tau \right) bu_d(k).
\] (11)

We next solve the integral in brackets by defining the variable \( s := kT + T - \tau \):

\[
\int_{kT}^{kT+T} e^{\lambda (kT+T-\tau)} d\tau = \int_0^T e^{\lambda s} (-ds) = \int_T^0 e^{\lambda s} ds = \frac{e^{\lambda T} - 1}{\lambda}.
\]

Substituting in (11) we conclude

\[
x_d(k + 1) = \lambda_d x_d(k) + b_d u_d(k)
\] (12)

where

\[
\lambda_d = e^{\lambda T}, \quad b_d = \frac{e^{\lambda T} - 1}{\lambda} b.
\]

Thus, (12) evaluates the state of the continuous-time model (10) at the next sample time. We refer to (12) as the ‘discretization’ of (10).