Lectures on Dynamic Systems and Control

Mohammed Dahleh Munther A. Dahleh George Verghese
Department of Electrical Engineering and Computer Science
Massachuasetts Institute of Technology[1]

Chapter 8

Simulation/Realization

8.1 Introduction

Given an nth-order state-space description of the form

$$\dot{x}(t) = f\left(x(t), u(t), t\right) \quad \text{(state evolution equations)} \tag{8.1}$$

$$y(t) = g\left(x(t), u(t), t\right) \quad \text{(instantaneous output equations)} \ . \tag{8.2}$$

(which may be CT or DT, depending on how we interpret the symbol $\dot{x}$), how do we *simulate* the model, i.e., how do we implement it or *realize* it in hardware or software? In the DT case, where $\dot{x}(t) = x(t+1)$, this is easy if we have available: (i) storage registers that can be updated at each time step (or "clock cycle") — these will store the state variables; and (ii) a means of evaluating the functions $f(\,\cdot\,)$ and $g(\,\cdot\,)$ that appear in the state-space description — in the linear case, all that we need for this are multipliers and adders. A straightforward realization is then obtained as shown in the figure below. The storage registers are labeled D for (one-step) *delay*, because the output of the block represents the data currently stored in the register while the input of such a block represents the data waiting to be read into the register at the *next* clock pulse. In the CT case, where $\dot{x}(t) = dx(t)/dt$, the only difference is that the delay elements are replaced by integrators. The outputs of the integrators are then the state variables.

8.2 Realization from I/O Representations

In this section, we will describe how a state space realization can be obtained for a causal input-output dynamic system described in terms of convolution.

8.2.1 Convolution with an Exponential

Consider a causal DT LTI system with impulse response $h[n]$ (which is 0 for $n < 0$):

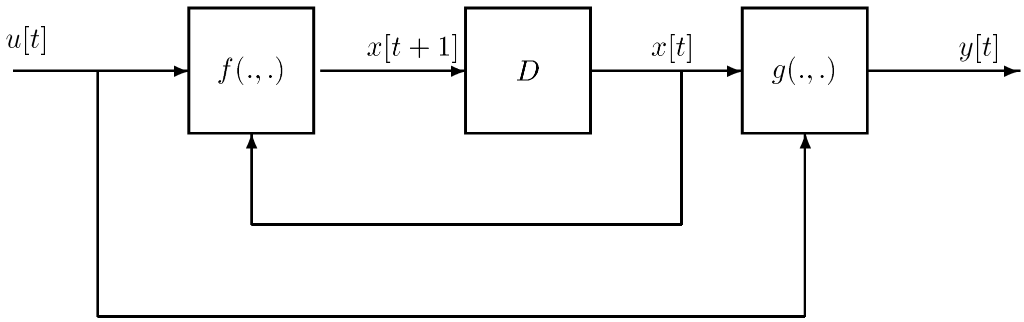


Figure 8.1: Simulation Diagram

$$\begin{aligned} y[n] &= \sum_{-\infty}^{n} h[n-k]u[k] \\ &= \Big(\sum_{-\infty}^{n-1} h[n-k]u[k] \Big) \; + \; h[0]u[n] \end{aligned} \tag{8.3}$$

The first term above, namely

$$x[n] = \sum_{-\infty}^{n-1} h[n-k]u[k] \tag{8.4}$$

represents the effect of the past on the present. This expression shows that, in general (i.e. if $h[n]$ has no special form), the number $x[n]$ has to be recomputed from scratch for each n. When we move from n to $n+1$, none of the past input, i.e. $u[k]$ for $k \leq n$, can be discarded, because all of the past will again be needed to compute $x[n+1]$. In other words, the memory of the system is infinite.

Now look at an instance where special structure in $h[n]$ makes the situation much better. Suppose

$$h[n] = \lambda^n \quad \text{for } n \geq 0, \text{ and } 0 \text{ otherwise} \tag{8.5}$$

Then

$$x[n] = \sum_{-\infty}^{n-1} \lambda^{n-k} u[k] \tag{8.6}$$

and

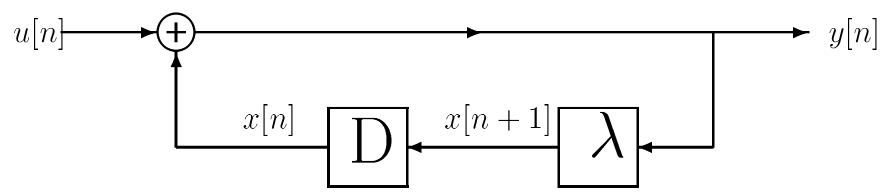
$$\begin{aligned} x[n+1] &= \sum_{-\infty}^{n} \lambda^{n+1-k} u[k] \\ &= \lambda \Big(\sum_{-\infty}^{n-1} \lambda^{n-k} u[k] \Big) \; + \; \lambda u[n] \\ &= \lambda x[n] + \lambda u[n] \end{aligned} \tag{8.7}$$

(You will find it instructive to graphically represent the convolutions that are involved here, in order to understand more visually why the relationship (8.7) holds.) Gathering (8.3) and (8.6) with (8.7), we obtain a pair of equations that together constitute a *state-space description* for this system:

$$x[n+1] = \lambda x[n] + \lambda u[n] \tag{8.8}$$

$$y[n] = x[n] + u[n] \tag{8.9}$$

To *realize* this model in hardware, or to *simulate* it, we can use a delay-adder-gain system that is obtained as follows. We start with a delay element, whose output will be $x[n]$ when its input is $x[n+1]$. Now the state evolution equation tells us how to combine the present output of the delay element, $x[n]$, with the present input to the system, $u[n]$, in order to obtain the present input to the delay element, $x[n+1]$. This leads to the following block diagram, in which we have used the output equation to determine how to obtain $y[n]$ from the present state and input of the system:

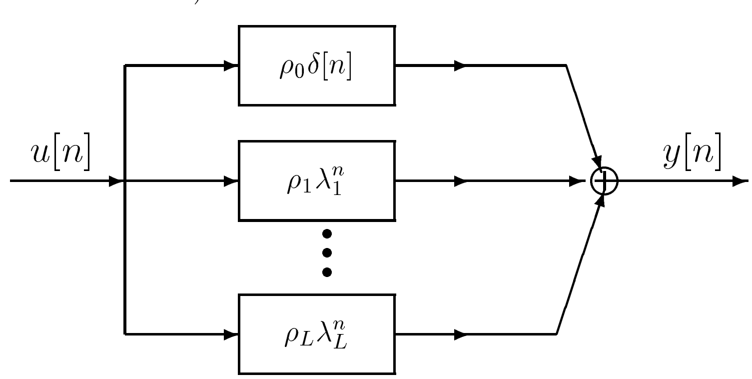


8.2.2 Convolution with a Sum of Exponentials

Consider a more complicated causal impulse response than the previous example, namely

$$h[n] = \rho_0 \delta[n] + (\, \rho_1 \lambda_1^n + \rho_2 \lambda_2^n + \cdots + \rho_L \lambda_L^n \,) \tag{8.10}$$

with the ρ_i being constants. The following block diagram shows that this system can be considered as being obtained through the parallel interconnection of causal subsystems that are as simple as the one treated earlier, plus a direct feedthrough of the input through the gain ρ_0 (each block is labeled with its impulse response, with causality implying that these responses are 0 for $n < 0$):



Motivated by the above structure and the treatment of the earlier, let us define a state variable for each of the L subsystems:

$$x_i[n] = \sum_{-\infty}^{n-1} \lambda_i^{n-k} u[k] \ , \quad i = 1, 2, \ldots, L \tag{8.11}$$

With this, we immediately obtain the following state-evolution equations for the subsystems:

$$x_i[n+1] = \lambda_i x_i[n] + \lambda_i u[n] \ , \quad i = 1, 2, \ldots, L \tag{8.12}$$

Also, after a little algebra, we directly find

$$y[n] = \rho_1 x_1[n] + \rho_2 x_2[n] + \cdots + \rho_L x_L[n] + (\sum_{0}^{L} \rho_i)\, u[n] \tag{8.13}$$

We have thus arrived at an Lth-order state-space description of the given system. To write the above state-space description in matrix form, define the state vector at time n to be

$$\mathbf{x}[n] = \begin{pmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_L[n] \end{pmatrix} \tag{8.14}$$

Also define the diagonal matrix $\mathbf{A}$, column vector $\mathbf{b}$, and row vector $\mathbf{c}$ as follows:

$$\mathbf{A} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \lambda_L \end{pmatrix} \ , \quad \mathbf{b} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_L \end{pmatrix} \tag{8.15}$$

$$\mathbf{c} = \begin{pmatrix} \rho_1 & \rho_2 & \cdots & \cdots & \cdots & \rho_L \end{pmatrix} \tag{8.16}$$

Then our state-space model takes the desired matrix form, as you can easily verify:

$$\mathbf{x}[n+1] = \mathbf{A}\mathbf{x}[n] + \mathbf{b}u[n] \tag{8.17}$$

$$y[n] = \mathbf{c}\mathbf{x}[n] + \mathtt{d}u[n] \tag{8.18}$$

where

$$\mathtt{d} = \sum_{0}^{L} \rho_i \tag{8.19}$$

8.3 Realization from an LTI Differential/Difference equation

In this section, we describe how a realization can be obtained from a difference or a differential equation. We begin with an example.

Example 8.1 (State-Space Models for an LTI Difference Equation)

Let us examine some ways of representing the following input-output difference equation in state-space form:

$$y[n] + a_1 y[n-1] + a_2 y[n-2] = b_1 u[n-1] + b_2 u[n-2] \tag{8.20}$$

For a first attempt, consider using as state vector the quantity

$$\mathbf{x}[n] = \begin{pmatrix} y[n-1] \\ y[n-2] \\ u[n-1] \\ u[n-2] \end{pmatrix} \tag{8.21}$$

The corresponding 4th-order state-space model would take the form

$$\begin{aligned} \mathbf{x}[n+1] = \begin{pmatrix} y[n] \\ y[n-1] \\ u[n] \\ u[n-1] \end{pmatrix} &= \begin{pmatrix} -a_1 & -a_2 & b_1 & b_2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} y[n-1] \\ y[n-2] \\ u[n-1] \\ u[n-2] \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u[n] \\ y[n] &= \begin{pmatrix} -a_1 & -a_2 & b_1 & b_2 \end{pmatrix} \begin{pmatrix} y[n-1] \\ y[n-2] \\ u[n-1] \\ u[n-2] \end{pmatrix} + (\, 0 \,)\, u[n] \end{aligned} \tag{8.22}$$

If we are somewhat more careful about our choice of state variables, it is possible to get more economical models. For a 3rd-order model, suppose we pick as state vector

$$\mathbf{x}[n] = \begin{pmatrix} y[n] \\ y[n-1] \\ u[n-1] \end{pmatrix} \tag{8.23}$$

The corresponding 3rd-order state-space model takes the form

$$\begin{aligned} \mathbf{x}[n+1] = \begin{pmatrix} y[n+1] \\ y[n] \\ u[n] \end{pmatrix} &= \begin{pmatrix} -a_1 & -a_2 & b_2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y[n] \\ y[n-1] \\ u[n-1] \end{pmatrix} + \begin{pmatrix} b_1 \\ 0 \\ 1 \end{pmatrix} u[n] \\ y[n] &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} y[n] \\ y[n-1] \\ u[n-1] \end{pmatrix} + (\, 0 \,)\, u[n] \end{aligned} \tag{8.24}$$

A still more clever/devious choice of state variables yields a 2nd-order state-space model. For this, pick

$$\mathbf{x}[n] = \begin{pmatrix} y[n] \\ -a_2 y[n-1] + b_2 u[n-1] \end{pmatrix} \tag{8.25}$$

The corresponding 2nd-order state-space model takes the form

$$\begin{aligned} \begin{pmatrix} y[n+1] \\ -a_2 y[n] + b_2 u[n] \end{pmatrix} &= \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix} \begin{pmatrix} y[n] \\ -a_2 y[n-1] + b_2 u[n-1] \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u[n] \\ y[n] &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} y[n] \\ -a_2 y[n-1] + b_2 u[n-1] \end{pmatrix} + (\,0\,)\, u[n] \end{aligned} \tag{8.26}$$

It turns out to be impossible in general to get a state-space description of order lower than 2 in this case. This should not be surprising, in view of the fact that we started with a 2nd-order difference equation, which we know (from earlier courses!) requires two initial conditions in order to solve forwards in time. Notice how, in each of the above cases, we have incorporated the information contained in the original difference equation that we started with.

This example was built around a second-order difference equation, but has natural generalizations to the nth-order case, and natural parallels in the case of CT differential equations.

Next, we will present two realizations of an nth-Order LTI differential equation. While realizations are not unique, these two have certain nice properties that will be discussed in the future.

8.3.1 Observability Canonical Form

Suppose we are given the LTI differential equation

$$y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_0 y = b_0 u + b_1 \dot{u} + \cdots + b_{n-1} u^{(n-1)},$$

which can be rearranged as

$$y^{(n)} = (b_{n-1} u^{(n-1)} - b_{n-1} y^{(n-1)}) + (b_{n-2} u^{(n-2)} - a_{n-2} y^{(n-2)}) + \cdots + (b_0 u - a_0 y).$$

Integrated n times, this becomes

$$y = \int (b_{n-1} u - a_{n-1} y) + \int\int (b_{n-2} u - a_{n-2} y) + \cdots + \underbrace{\int \cdots \int}_{n} (b_0 u - a_0 y). \tag{8.27}$$

The block diagram given in Figure 8.2 then follows directly from (8.27). This particular realization is called the *observability canonical form* realization — “canonical” in the sense of

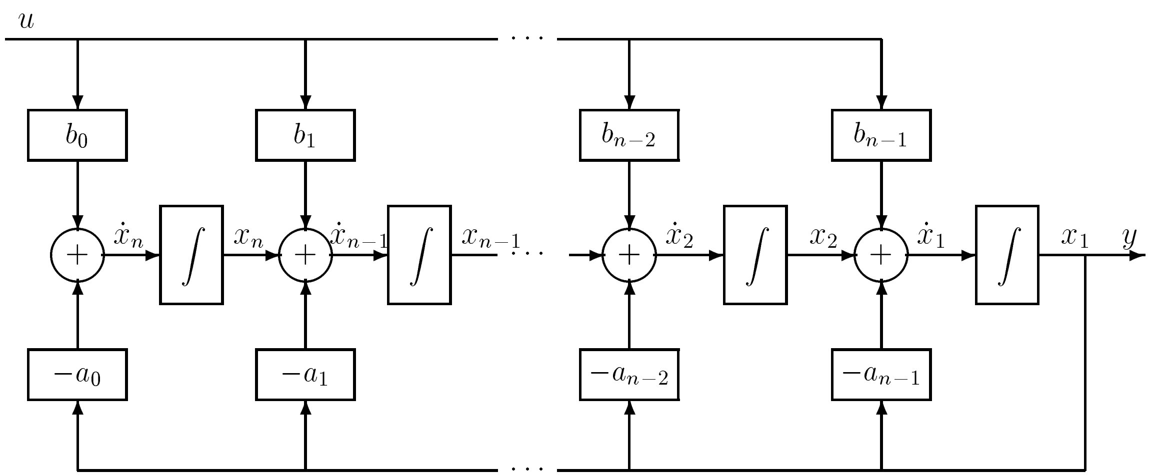


Figure 8.2: Observability Canonical Form

"simple" (but there is actually a strict mathematical definition as well), and "observability" for reasons that will emerge later in the course.

We can now read the state equations directly from Figure 8.2, once we recognize that the natural state variables are the outputs of the integrators:

$$\begin{aligned}
\dot{x}_1 &= -a_{n-1}x_1 + x_2 + b_{n-1}u \\
\dot{x}_2 &= -a_{n-2}x_1 + x_3 + b_{n-2}u \\
&\vdots \\
\dot{x}_n &= -a_0 x_1 + b_0 u \\
\\
y &= x_1 .
\end{aligned}$$

If this is written in our usual matrix form, we would have

$$A = \begin{bmatrix} -a_{n-1} & 1 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ & & & & 1 \\ -a_0 & 0 & & \cdots & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} b_{n-1} \\ b_{n-2} \\ \vdots \\ \vdots \\ b_0 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}.$$

The matrix A is said to be in *companion form*, a term used to refer to any one of four matrices whose pattern of 0's and 1's is, or resembles, the pattern seen above. The characteristic polynomial of such a matrix can be directly read off from the remaining coefficients, as we shall see when we talk about these polynomials, so this matrix is a "companion" to its characteristic polynomial.

8.3.2 Reachability Canonical Form

There is a "dual" realization to the one presented in the previous section for the LTI differential equation

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_0 y = c_0 u + c_1 \dot{u} + \cdots + c_{n-1}u^{(n-1)}. \tag{8.28}$$

First, consider a special case of this, namely the differential equation

$$w^{(n)} + a_{n-1}w^{(n-1)} + \cdots + a_0 w = u \tag{8.29}$$

To obtain an nth-order state-space realization of the system in 8.29, define

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} w \\ \dot{w} \\ \ddot{w} \\ \vdots \\ \frac{d^{n-2}w}{dt^{n-2}} \\ \frac{d^{n-1}w}{dt^{n-1}} \end{bmatrix}.$$

Then it is easy to verify that the following state-space description represents the given model:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \vdots & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -a_0(t) & -a_1(t) & -a_2(t) & \ldots & -a_{n-2}(t) & -a_{n-1}(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$w = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}.$$

(The matrix A here is again in one of the companion forms; the two remaining companion forms are the transposes of the one here and the transpose of the one in the previous section.) Suppose now that we want to realize another special case, namely the differential equation

$$r^{(n)} + a_{n-1}r^{(n-1)} + \cdots + a_0 r = \dot{u} \tag{8.30}$$

which is the same equation as (8.29), except that the RHS is $\dot{u}$ rather than u. By linearity, the response of (8.30) will $r = \dot{w}(t)$, and this response can be obtained from the above realization by simply taking the output to be x_2 rather than x_1, since $x_2 = \dot{w} = r$.

Superposing special cases of the preceding form, we see that if we have the differential equation (8.28), with an RHS of

$$c_0 u + c_1 \dot{u} + \cdots + c_{n-1} u^{(n-1)}$$

then the above realization suffices, provided we take the output to be

$$y = c_0 x_1 + c_1 x_2 + \cdots + c_{n-1} x_n. \tag{8.31}$$

i.e., we just change the output equation to have

$$c = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \end{bmatrix}. \tag{8.32}$$

A block diagram of the final realization is shown below in 8.3. This is called the *reachability or controllability canonical form.*

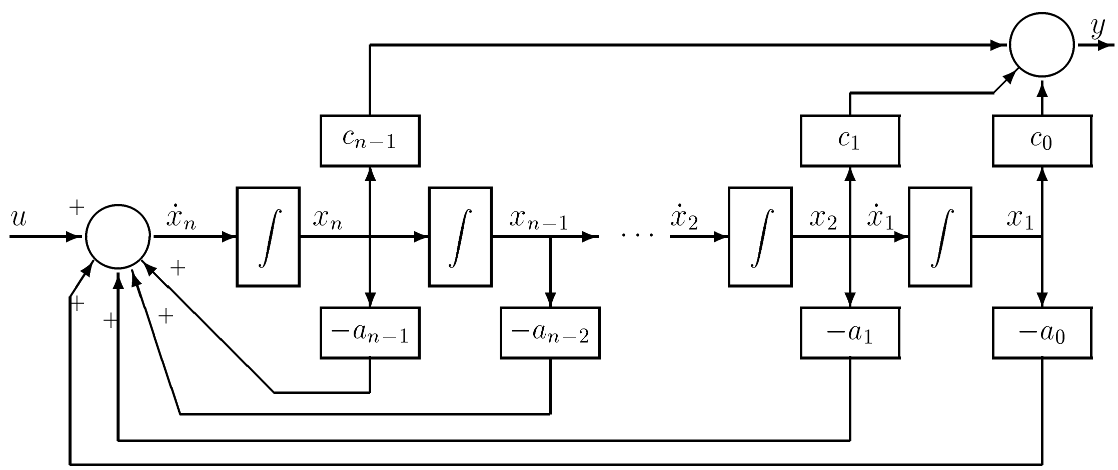


Figure 8.3: Reachability Canonical Form

Finally, for the obvious DT difference equation that is analogous to the CT differential equation that we used in this example, the same scheme will work, with derivatives replaced by differences.

Exercises

Exercise 8.1 Suppose we wish to realize a *two-input* differential equation of the form

$$\begin{aligned} y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_0 y = \quad & b_{01}u_1 + b_{11}\dot{u}_1 + \cdots + b_{n-1,1}u_1^{(n-1)} \\ + \quad & b_{02}u_2 + b_{12}\dot{u}_2 + \cdots + b_{n-1,2}u_2^{(n-1)} \end{aligned}$$

Show how you would modify the observability canonical realization to accomplish this, still using only n integrators.

Exercise 8.2 How would reachability canonical realization be modified if the linear differential equation that we started with was time varying rather than time invariant?

Exercise 8.3 Show how to modify the reachability canonical realization— but still using only n integrators — to obtain a realization of a *two-output* system of the form

$$\begin{aligned} y_1^{(n)} + a_{n-1}y_1^{(n-1)} + \cdots + a_0 y_1 &= c_{10}u + c_{11}\dot{u} + \cdots + c_{1,n-1}u^{(n-1)} \ , \\ y_2^{(n)} + a_{n-1}y_2^{(n-1)} + \cdots + a_0 y_2 &= c_{20}u + c_{21}\dot{u} + \cdots + c_{2,n-1}u^{(n-1)} . \end{aligned}$$

Exercise 8.4 Consider the two-input two-output system:

$$\begin{aligned} \dot{y}_1 &= y_1 + \alpha u_1 + u_2 \ , \\ \dot{y}_2 &= y_2 + u_1 + u_2 \end{aligned}$$

(a) Find a realization with the minimum number of states when $\alpha \neq 1$.

(b) Find a realization with the minimum number of states when $\alpha = 1$.

6.241J / 16.338J Dynamic Systems and Control
Spring 2011