Chapter 8: State Representations of Systems

In this chapter we introduce the concept of a state vector and see how to express systems in their state-space form. We then develop methods for obtaining solutions for the outputs of systems given in state-space form.

Recall that linear time-invariant (LTI) discrete-time systems can be expressed in their **input-output form** with a difference equation

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = c_0 v[n-N], \quad \text{for } n \geq 0$$

An alternative equivalent form is

$$y[n+N] + \sum_{i=0}^{N-1} b_i y[n+i] = c_0 v[n], \quad \text{for } n \geq -N$$

where $v[n]$ and $y[n]$ are the inputs and outputs, and the coefficients a_i and c_0 are constants.
Similarly an LTI continuous-time system can expressed as a differential equation

$$\frac{dy^N}{dt^N} + a_{N-1}\frac{dy^{N-1}}{dt^{N-1}} + \ldots a_1 \frac{dy}{dt} + a_0 y(t) = c_0 v(t) \quad \text{for } t \geq 0$$

Definition

A **state vector** for a system is a vector of the form

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}, \quad \text{or} \quad x[n] = \begin{bmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_N[n] \end{bmatrix}$$

- Together with a known input $v(t)$, the state vector contains sufficient information at any time t (or n) to enable the future outputs $y(t)$ (or $y[n]$) to be uniquely determined.
- The variables $x_1, \ldots, x_N$ are called the **state variables** of the system, and N is the **dimension** of the state.
- The trace $\{x(t) : t \in \mathbf{R}\}$ (or $\{x[n] : n \in \mathbf{Z}\}$) is the **state trajectory** of the system.

Definition

The **state equations** for a linear time-invariant system with an N-dimensional state vector are

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \quad \text{or} \quad \begin{aligned} x[n+1] &= Ax[n] + Bv[n] \\ y[n] &= Cx[n] + Dv[n] \end{aligned}$$

where

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_N(t) \end{bmatrix}, \quad \text{and} \quad x[n+1] = \begin{bmatrix} x_1[n+1] \\ x_2[n+1] \\ \vdots \\ x_N[n+1] \end{bmatrix}$$

and A is $N \times N$ matrix, B is a $N \times 1$ vector, C is a $1 \times N$ vector, and D is a scalar.

To obtain a state space model for the continuous-time system

$$\frac{dy^N}{dt^N} + a_{N-1}\frac{dy^{N-1}}{dt^{N-1}} + \ldots a_1\frac{dy}{dt} + a_0 y(t) = c_0 v(t) \quad \text{for } t \geq 0$$

1. Define a state vector $x(t)$ with state variables

$$x_1(t) = \frac{1}{c_0}y(t),\ x_2(t) = \dot{x}_1(t),\ x_3(t) = \dot{x}_2(t), \ldots, x_N(t) = \dot{x}_{N-1}(t)$$

2. Then the state equations are

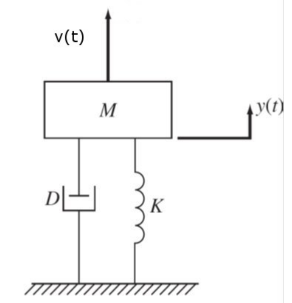
$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
&\vdots \\
\dot{x}_N(t) &= -a_0 x_1(t) - a_1 x_2(t) - \cdots - a_{N-1} x_N(t) + v(t) \\
y(t) &= c_0 x_1(t)
\end{aligned}$$

If we write these in matrix form we obtain the **state space representation**

$$
\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{N-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} v(t) \\
y(t) &= \begin{bmatrix} c_0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}
\end{aligned}
$$

This is known as the **controller canonical form**. It is not unique; later we will see other ways to achieve state space representations.

Example (State space model for the Mass Spring Damper: I)



$$\frac{d^2y}{dt^2} + \frac{D}{M}\frac{dy}{dt} + \frac{K}{M}y(t) = v(t)$$

The system is second-order, so to obtain a state space model we define a state vector $x(t)$ with two state variables

$$x_1(t) = y(t), \quad x_2(t) = \dot{y}(t)$$

Example (State space model for the Mass Spring Damper: II)

Then the state equations are

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -\frac{K}{M}x_1(t) - \frac{D}{M}x_2(t) + v(t) \\
y(t) &= x_1(t)
\end{aligned}
$$

and writing these in matrix form we obtain the **state space representation**

$$
\begin{aligned}
\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{D}{M} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t) \\
y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}
\end{aligned}
$$

Similarly, to obtain a state space model for the discrete-time system

$$y[n+N]+\sum_{i=0}^{N-1} a_i y[n+i] = c_0 v[n], \quad \text{for } n \geq -N$$

1. Define a state vector $x[n]$ with state variables

$$x_1[n] = \frac{1}{c_0} y[n],\ x_2[n] = y[n+1],\ \ldots, x_N[n] = y[n+N-1]$$

2. Then the state equations are

$$\begin{aligned} x_1[n+1] &= x_2[n] \\ x_2[n+1] &= x_3[n] \\ &\vdots \\ x_N[n+1] &= -a_0 x_1[n] - a_1 x_2[n] - \cdots - a_{N-1} x_N[n] + v[n] \\ y[n] &= c_0 x_1[n] \end{aligned}$$

The state matrices are the same as for continuous-time systems.

Definition (Multiple-input Multiple output systems)

A continuous-time system in state space form

$$\begin{aligned}\dot{x}(t) &= Ax(t)+Bv(t)\\ y(t) &= Cx(t)+Dv(t)\end{aligned}$$

is a **Multiple-input Multiple output system**, or **MIMO system**, if A is an $N\times N$ matrix, B is a $N\times m$ matrix, C is a $p\times N$ matrix and D is a $p\times m$ matrix, where N, m, and p are all integers greater than or equal to 2. Systems where $p=m=1$ are called **Scalar-input Scalar output systems**, or **SISO** systems.

MIMO systems can have more than one input variable, and more than one output variable. The state space equations for a discrete-time MIMO system are very similar:

$$\begin{aligned}x[n+1] &= Ax[n]+Bv[n]\\ y[n] &= Cx[n]+Dv[n]\end{aligned}$$

Example (Coupled two-car MIMO system: I)

Two cars driving along a level surface may be described by the equations

$$\begin{aligned} \ddot{d}_1(t) + \frac{k_f}{M}\dot{d}_1(t) &= \frac{1}{M} f_1(t) \\ \ddot{d}_2(t) + \frac{k_f}{M}\dot{d}_2(t) &= \frac{1}{M} f_2(t) \\ w(t) &= d_2(t) - d_1(t) \end{aligned}$$

where d_1 and d_2 are the positions of the first and second cars, and M and k_f are constants.

Example (Coupled two-car MIMO system: II)

If we introduce state variables

$$x_1(t) = \dot{d}_1(t), \quad x_2(t) = \dot{d}_2(t), \quad x_3(t) = w(t)$$

and take

$$y(t) = \begin{bmatrix} \dot{d}_1(t) \\ w(t) \end{bmatrix}$$

as the output, then the state space model is

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} \frac{-k_f}{M} & 0 & 0 \\ 0 & -\frac{K}{M} & 0 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{M} & 0 \\ 0 & \frac{1}{M} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \end{aligned}$$

Definition (Integrator Realizations)

An **integrator realization** is a diagram that represents a continuous-time state-space system in terms of integrators, summers and scalar multipliers.

The steps of the realization process are as follows

1. For each state variable x_i, construct an integrator and define the output of the integrator to be x_i. Hence the input to the integrator is $\dot{x}_i$. A system with N states will need N integrators.
2. Put a summer in front of each integrator. Feed into the summer scalar multiples of the state variables according to the i-th state equation. $\dot{x}_i(t) = A_i x(t) + B_i v(t)$, where A_i and B_i are the i-th rows of of A and B, respectively.
3. Put scalar multiples of the state variables into a summer to realize the output equation $y_i(t) = C_i(t) + D_i v(t)$, where C_i and D_i are the i-th rows of of C and D, respectively.

Example (Integrator Realization I)

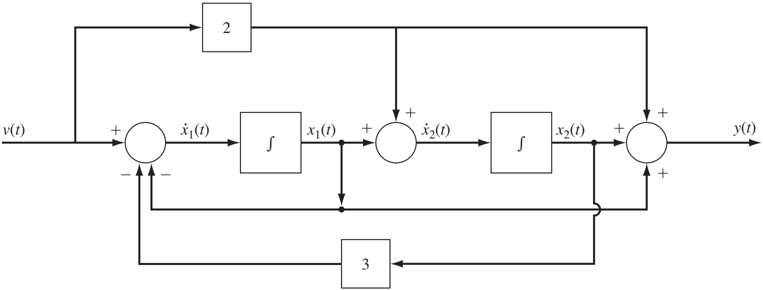
Draw the integrator realization for the system

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) - 3x_2(t) + v(t)\\ \dot{x}_2(t) &= x_1(t) + 2v(t)\\ y(t) &= x_1(t) + x_2(t) + 2v(t)\end{aligned}$$

1. Integrators are drawn as a box containing the $\int$ symbol. We draw one arrow entering the box (for $\dot{x}_i$) and one arrow leaving the box (for x_i).
2. A summer is drawn as a circle with arrows entering it, with $+/-$ used to indicate add/subtract.
3. Scalar multipliers are drawn as a box with the scalar inside.
4. The input $v(t)$ is drawn as an arrow on the left of the diagram, and the output $y(t)$ is an arrow on the right of the diagram.

Example (Integrator Realization II)

- The summer on the left creates the state equation $\dot{x}_1(t) = -x_1(t) - 3x_2(t) + v(t)$
- The middle summer creates the state equation $\dot{x}_2(t) = x_1(t) + 2v(t)$.
- The summer on the right creates the output equation $y(t) = x_1(t) + x_2(t) + 2v(t)$



Definition (Unit Delay Realizations)

A **unit delay realization** is a diagram that represents a discrete-time state-space system in terms of unit delays, summers and scalar multipliers.

The process for drawing the diagram is the same as for building integrator realizations for continuous-time systems, except that unit delays are used in place of integrators.

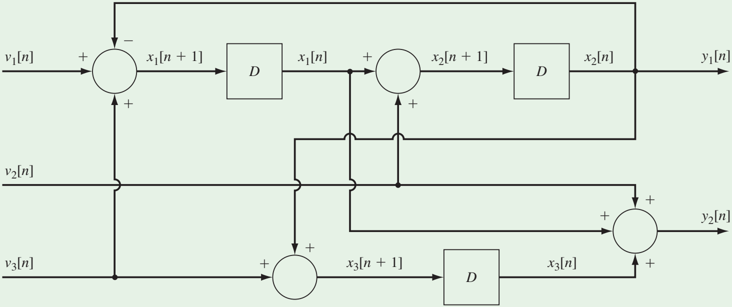
Example (Unit Delay Realization I)

Draw the unit delay realization for the third-order system

$$\begin{aligned} x_1[n+1] &= -x_2[n] + v_1[n] + v_3[n] \\ x_2[n+1] &= x_1[n] + v_2[n] \\ x_3[n+1] &= x_2[n] + v_3[n] \\ y_1[n] &= x_2[n] \\ y_2[n] &= x_1[n] + x_3[n] + v_2[n] \end{aligned}$$

Example (Unit Delay Realization II)

1. We create a unit delay block containing the D symbol for each variable, with $x_i[n+1]$ entering and $x_i[n]$ leaving the block.
2. We have three input variables v_1, v_2 and v_3 drawn on the left, and two output variables y_1 and y_2 drawn on the right side of the diagram.
3. Summers are used to implement each state equation.



Definition (Matrix Exponential)

Let A be any real $N \times N$ matrix. For any $t \in \mathbf{R}$, the **matrix exponential** of A is the matrix function defined by the matrix power series

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + \frac{A^3 t^3}{3!} + \frac{A^4 t^4}{4!} + \dots$$

where I is the $N \times N$ identity matrix.

Theorem (Properties of the Matrix Exponential)

- *For any two real numbers t and s,*

$$e^{A(t+s)} = e^{At} e^{As}$$

- *The matrix inverse of e^{At} is e^{-At}. The time derivative of e^{At} is given by*

$$\frac{d}{dt} e^{At} = A e^{At} = e^{At} A$$

Theorem (Matrix Exponential of a Diagonal Matrix)

Let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be real or complex numbers and let Λ be the diagonal matrix

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \ldots & 0 & \lambda_N \end{bmatrix}$$

Then the matrix exponentials of Λ are, for $t \in \mathbf{R}$ and integer n,

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \ldots & 0 \\ 0 & e^{\lambda_2 t} & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \ldots & 0 & e^{\lambda_N t} \end{bmatrix}, \quad \Lambda^n = \begin{bmatrix} {\lambda_1}^n & 0 & \ldots & 0 \\ 0 & {\lambda_2}^n & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \ldots & 0 & {\lambda_N}^n \end{bmatrix}$$

Theorem (Eigenvalues of similar matrices)

Let A and $\bar{A}$ be square matrices, and assume there exists an invertible matrix P such that

$$\bar{A} = PAP^{-1}$$

Then A and $\bar{A}$ have the same eigenvalues. We say that A and $\bar{A}$ are **similar matrices***.*

Definition (Diagonalizability)

For any $N \times N$ matrix A, we say that A is **diagonalizable** if there exists an invertible matrix P and a diagonal matrix Λ such that

$$\Lambda = PAP^{-1}$$

Theorem

Assume A has N distinct eigenvalues $\{\lambda_1, \dots, \lambda_N\}$ and let $\{v_1, \dots, v_N\}$ be the corresponding eigenvectors. Then A is diagonalizable with

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_N \end{bmatrix}, \quad P = [v_1 \dots v_N]^{-1}$$

- Note that Λ and P will be complex matrices if A has complex eigenvalues.
- As $\Lambda = PAP^{-1}$, we see that Λ and A are similar matrices, and hence they have the same eigenvalues.

For diagonalizable matrices, there is a convenient way to compute their matrix exponentials.

Theorem

Let A be a diagonalizable matrix, and let P and Λ be such that

$$\Lambda = PAP^{-1}$$

with P an invertible matrix and Λ a diagonal matrix. Then for any $t \in \mathbf{R}$,

$$e^{At} = P^{-1} e^{\Lambda t} P$$

and for any integer n,

$$A^n = P^{-1} \Lambda^n P$$

Example

Find e^{At} if

$$A = \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix}$$

Since A is an upper triangular matrix, the eigenvalues appear on the leading diagonal: $\lambda_1 = 0$ and $\lambda_2 = 1$. We find that the corresponding eigenvectors are $v_1 = [1\ 0]^T$, $v_2 = [-1\ 1]^T$. Hence a diagonalization for A is given by

$$\Lambda = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Then $A = P^{-1}\Lambda P$ and

$$\begin{aligned} e^{At} &= P^{-1}e^{\Lambda t}P \\ &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^t \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1-e^t \\ 0 & e^t \end{bmatrix} \end{aligned}$$

Theorem

The continuous-time system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned}$$

has solution

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\lambda)}Bv(\lambda)\,d\lambda, \quad \text{for } t \geq 0$$

and output

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\lambda)}Bv(\lambda)\,d\lambda + Dv(t), \quad \text{for } t \geq 0$$

The exponential matrix e^{At} is known as the **state-transition matrix** *of the system.*

Definition

When the zero input ($v(t) = 0$) is applied to a continuous-time system, the solution and output are

$$x_{zi}(t) = e^{At}x(0), \quad y_{zi}(t) = Ce^{At}x(0), \quad \text{for } t \geq 0$$

They are called the **zero-input solution** (or **unforced solution**) and the **zero-input response** of the system.
When the initial state is zero ($x(0) = 0$), the solution and output are

$$\begin{aligned} x_{zs}(t) &= \int_0^t e^{A(t-\lambda)}Bv(\lambda)\, d\lambda, \\ y_{zs}(t) &= \int_0^t Ce^{A(t-\lambda)}Bv(\lambda)\, d\lambda + Dv(t), \text{ for } t \geq 0 \end{aligned}$$

They are called the **zero-state solution** and the **zero-state response** of the system.

Theorem

The discrete-time system

$$\begin{aligned} x[n+1] &= Ax[n] + Bv[n] \\ y[n] &= Cx[n] + Dv[n] \end{aligned}$$

has solution

$$x[n] = A^n x[0] + \sum_{i=0}^{n-1} A^{n-i-1} Bv[i], \quad \textit{for } n \geq 1$$

and output

$$y[n] = CA^n x[0] + \sum_{i=0}^{n-1} CA^{n-i-1} Bv[i] + Dv[n], \quad \textit{for } n \geq 1$$

The matrix A^n is known as the **state-transition matrix** *of the system.*

Definition

When the zero input ($v[n] = 0$) is applied to a discrete-time system, we obtain the **zero-input solution** and the **zero-input response** of the system:

$$x_{zi}[n] = A^n x[0], \quad y_{zi}[n] = CA^n x[0], \quad \text{for } n \geq 1$$

When the initial state is zero ($x[0] = 0$), we obtain the **zero-state output** and the **zero-state response** of the system:

$$\begin{aligned} x_{zs}[n] &= \sum_{i=0}^{n-1} A^{n-i-1} Bv[i], \quad \text{for } n \geq 1 \\ y_{zs}[n] &= \begin{cases} Dv[0], & n = 0 \\ \sum_{i=0}^{n-1} CA^{n-i-1} Bv[i] + Dv[n], & n > 0 \end{cases} \end{aligned}$$

Definition (Coordinate transformations)

Let $x(t)$ be a state vector in $\mathbf{R}^n$ and let P be an invertible $n \times n$ matrix A. Then we can define a new state vector $\bar{x}(t)$ by

$$\bar{x}(t) = Px(t)$$

We say that P is a **coordinate transformation** matrix.

Definition (Equivalent state representations)

Let (A, B, C, D) be the state matrices of either a continuous-time or discrete-time system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \quad \text{or} \quad \begin{aligned} x[n+1] &= Ax[n] + Bv[n] \\ y[n] &= Cx[n] + Dv[n] \end{aligned}$$

and let $\bar{x} = Px$, for some coordinate transformation P.

Definition (Equivalent state representations)

Introduce state matrices $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ where

$$\bar{A} = PAP^{-1},\ \bar{B} = PB,\ \bar{C} = CP^{-1}, \bar{D} = D$$

Then the state representation

$$\begin{aligned} \dot{\bar{x}}(t) &= \bar{A}\bar{x}(t) + \bar{B}v(t) \\ y(t) &= \bar{C}\bar{x}(t) + \bar{D}v(t) \end{aligned} \quad \text{or} \quad \begin{aligned} x[n+1] &= \bar{A}\bar{x}[n] + \bar{B}v[n] \\ y[n] &= \bar{C}\bar{x}[n] + \bar{D}v[n] \end{aligned}$$

is **equivalent** to the state representation

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \quad \text{or} \quad \begin{aligned} x[n+1] &= Ax[n] + Bv[n] \\ y[n] &= Cx[n] + Dv[n] \end{aligned}$$

We say that the quadruples of state matrices (A, B, C, D) and $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ are equivalent.

Theorem (Equivalent state representations I)

Let (A, B, C, D) and $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ be equivalent state space representations for an LTI system. For any input signal v, the output y can be obtained using either (A, B, C, D) or $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$.

Example (Equivalent state representations)

Consider the system with state space representation

$$\begin{aligned} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} -3 & 2 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} v(t) \\ y(t) &= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \end{aligned}$$

Introduce

$$P = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \bar{x}(t) = Px$$

Note that $\det(P) = 1$, so P is invertible.

Example (Equivalent state representations)

Introduce new state matrices

$$\bar{A} = PAP^{-1} = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix}, \; \bar{B} = PB = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; \bar{C} = CP^{-1} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Then the state space representation

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_1(t) \\ \dot{\bar{x}}_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} \end{aligned}$$

is an equivalent state representation for the system.

Observe the system is in now controller canonical form. Next we see how coordinate transformations can convert systems into a useful diagonal form.

Example (Diagonal state representations I)

Consider the system with state space representation

$$\begin{aligned}\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} v(t)\\ y(t) &= \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}\end{aligned}$$

Simple computations reveal that $\lambda_1 = -1$ and $\lambda_2 = -2$ are the eigenvalues of A, with corresponding eigenvectors $v_1 = [1 \ -1]^T$ and $v_2 = [1 \ -2]^T$. Hence the matrix A has distinct eigenvalues and is diagonalizable. The diagonal matrix Λ and coordinate transformation P are

$$\Lambda = \begin{bmatrix}-1 & 0\\ 0 & -2\end{bmatrix}, \quad P = \begin{bmatrix}1 & 1\\ -1 & -2\end{bmatrix}^{-1} = \begin{bmatrix}2 & 1\\ -1 & -1\end{bmatrix}$$

Example (Diagonal state representations II)

So we may use P to obtain an alternative state representation for the system in which A is replaced by a diagonal matrix. We define new coordinates $\bar{x} = Px$ and new state matrices

$$\bar{A} = PAP^{-1} = \Lambda,\ \bar{B} = PB = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \bar{C} = CP^{-1} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Then the diagonal state representation in $\bar{x}$-coordinates is

$$\begin{aligned} \begin{bmatrix} \dot{\bar{x}}_1(t) \\ \dot{\bar{x}}_2(t) \end{bmatrix} &= \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} v(t) \\ y(t) &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} \end{aligned}$$

is an equivalent state representation for the system.

Example (Diagonal state representations III)

Suppose the initial state is $x(0) = [2 - 2]^T$. Then

$$\bar{x}(0) = Px(0) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

The zero-input solution in $\bar{x}$-coordinates is

$$\begin{aligned} \bar{x}_{zi}(t) &= e^{\Lambda t}\bar{x}(0) \\ &= \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2e^{-t} \\ 0 \end{bmatrix} \end{aligned}$$

Example (Diagonal state representations IV)

Next suppose the input $v(t) = u(t)$, the step input. Then zero-state solution in $\bar{x}$-coordinates is

$$\begin{aligned}
\bar{x}_{zs}(t) &= \int_0^t e^{\Lambda(t-\lambda)}\bar{B}v(\lambda)\,d\lambda \\
&= \int_0^t \begin{bmatrix} e^{-(t-\lambda)} & 0 \\ 0 & e^{-2(t-\lambda)} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(\lambda)\,d\lambda \\
&= \int_0^t \begin{bmatrix} e^{-(t-\lambda)} \\ -e^{-2(t-\lambda)} \end{bmatrix} u(\lambda)\,d\lambda \\
&= \begin{bmatrix} 1 - e^{-t} \\ \frac{-1}{2}(1 - e^{-2t}) \end{bmatrix}
\end{aligned}$$

Example (Diagonal state representations V)

Hence the zero-input response and the zero-state response are

$$\begin{aligned}
y_{zi}(t) &= \bar{C}\bar{x}_{zi}(t) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2e^{-2t} \\ 0 \end{bmatrix} \\
&= 2e^{-2t} \\
y_{zs}(t) &= \bar{C}\bar{x}_{zs}(t) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 - e^{-t} \\ \frac{-1}{2}(1 - e^{-2t}) \end{bmatrix} \\
&= \frac{1}{2} - e^{-t} + \frac{1}{2}e^{-2t}
\end{aligned}$$

Thus the system output from the given initial value $x(0)$ is

$$y(t) = y_{zi}(t) + y_{zs}(t) = \frac{1}{2} - e^{-t} + \frac{5}{2}e^{-2t}$$