ELEC4410

Control Systems Design

Lecture 12: State Space Equivalence and Realisations

School of Electrical Engineering and Computer Science
The University of Newcastle

Outline

- Brief Review on Linear Algebra
- Equivalent State Equations
- Canonical Forms
- Realisations

Brief Review on Linear Algebra

Eigenvalues and Eigenvectors of a Matrix. They play a key role in the study of LTI systems and state equations.

A number $\lambda \in \mathbb{C}$ is an eigenvalue of a matrix $A \in \mathbb{R}^{n\times n}$ if there exists a nonzero vector $v \in \mathbb{R}^n$ such that

$$Av = \lambda v.$$

The vector v is a (right) eigenvector of A associated with the eigenvalue λ.

Brief Review on Linear Algebra

Eigenvalues and Eigenvectors of a Matrix. They play a key role in the study of LTI systems and state equations.

A number $\lambda \in \mathbb{C}$ is an eigenvalue of a matrix $A \in \mathbb{R}^{n \times n}$ if there exists a nonzero vector $v \in \mathbb{R}^n$ such that

$$Av = \lambda v.$$

The vector v is a (right) eigenvector of A associated with the eigenvalue λ.

Eigenvalues are found by solving the algebraic equation

$$(\lambda \mathbf{I} - A)v = 0.$$

This equation has nonzero solutions if the matrix $(\lambda \mathbf{I} - A)$ is singular (its determinant is zero).

Brief Review on Linear Algebra

Characteristic Polynomial of a Matrix

The characteristic polynomial of a matrix A is

$$\begin{aligned}\Delta(\lambda) &= \mathbf{det}(\lambda I - A) \\ &= \lambda^n + \alpha_1\lambda^{n-1} + \alpha_2\lambda^{n-2} + \cdots + \alpha_n.\end{aligned}$$

It is a *monic* polynomial (its leading coefficient is 1) of degree n with n real coefficients.

Brief Review on Linear Algebra

Characteristic Polynomial of a Matrix

The characteristic polynomial of a matrix A is

$$\begin{aligned}\Delta(\lambda) &= \det(\lambda I - A)\\ &= \lambda^n + \alpha_1\lambda^{n-1} + \alpha_2\lambda^{n-2} + \cdots + \alpha_n.\end{aligned}$$

It is a *monic* polynomial (its leading coefficient is 1) of degree n with n real coefficients.

Because for every root of $\Delta(\lambda)$ the matrix $(sI - A)$ is singular, we conclude that every root of $\Delta(\lambda)$ is an eigenvalue of A. Because a polynomial of degree n has n roots, a square matrix A has n eigenvalues (although not all necessarily different).

Brief Review on Linear Algebra

Characteristic Polynomial of a Matrix

The characteristic polynomial of a matrix A is

$$\begin{aligned}\Delta(\lambda) &= \det(\lambda I - A) \\ &= \lambda^n + \alpha_1\lambda^{n-1} + \alpha_2\lambda^{n-2} + \cdots + \alpha_n.\end{aligned}$$

It is a *monic* polynomial (its leading coefficient is 1) of degree n with n real coefficients.

Because for every root of $\Delta(\lambda)$ the matrix $(sI - A)$ is singular, we conclude that every root of $\Delta(\lambda)$ is an eigenvalue of A. Because a polynomial of degree n has n roots, a square matrix A has n eigenvalues (although not all necessarily different).

In MATLAB eigenvalues are computed with the function `r=eig(A)`; and the characteristic polynomial can be computed with the function `poly(A)`.

Brief Review on Linear Algebra

***Companion Form* Matrices.** To obtain the characteristic polynomial we need to expand **det**$(\lambda I - A)$. However, for some matrices the characteristic polynomial is evident.

One group of such matrices is that of companion form matrices

$$\begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 0 & 0 & 0 & -\alpha_4 \\ 1 & 0 & 0 & -\alpha_3 \\ 0 & 1 & 0 & -\alpha_2 \\ 0 & 0 & 1 & -\alpha_1 \end{bmatrix}$$

(and their transposes). They have the characteristic polynomial

$$\Delta(\lambda) = \lambda^4 + \alpha_1\lambda^3 + \alpha_2\lambda^2 + \alpha_3\lambda + \alpha_4 .$$

In MATLAB the command `compan(P)` forms a companion matrix with characteristic polynomial `P`.

Brief Review on Linear Algebra

***Diagonal* and *Jordan Form* Matrices.** Another case in which the characteristic polynomial is easily obtained is that in which the matrix is in diagonal form. For example,

$$\begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

has the characteristic polynomial

$$\Delta(\lambda) = (\lambda - \lambda_1) \times (\lambda - \lambda_2) \times \cdots \times (\lambda - \lambda_n)$$

Brief Review on Linear Algebra

If a matrix $\mathbf{A}$ is **diagonalisable**, it can always be taken to a diagonal form, $\bar{\mathbf{A}}$ say, by a similarity transformation $\bar{\mathbf{A}} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}$.

However, a matrix is not always diagonalisable. It depends on two cases

1. eigenvalues of $\mathbf{A}$ are all distinct
2. eigenvalues of $\mathbf{A}$ are not all distinct

We next analyse each case.

Brief Review on Linear Algebra

Eigenvalues of A are all distinct. In this case the set of associated eigenvectors, say $\{v_1, v_2, \ldots, v_n\}$, are **linearly independent**. This means that the matrix

$$Q = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

is nonsingular.

Brief Review on Linear Algebra

Eigenvalues of A are all distinct. In this case the set of associated eigenvectors, say $\{v_1, v_2, \ldots, v_n\}$, are **linearly independent**. This means that the matrix

$$Q = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

is nonsingular. Then, from the definition of eigenvalues,

$$\begin{aligned} AQ &= A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \\ &= \begin{bmatrix} Av_1 & Av_2 & \cdots & Av_n \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_n v_n \end{bmatrix} = Q\bar{A} \Leftrightarrow \bar{A} = Q^{-1}AQ. \end{aligned}$$

Brief Review on Linear Algebra

Eigenvalues of A are all distinct. In this case the set of associated eigenvectors, say $\{v_1, v_2, \dots, v_n\}$, are **linearly independent**. This means that the matrix

$$Q = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

is nonsingular. Then, from the definition of eigenvalues,

$$\begin{aligned} AQ &= A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \\ &= \begin{bmatrix} Av_1 & Av_2 & \cdots & Av_n \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_n v_n \end{bmatrix} = Q\bar{A} \Leftrightarrow \bar{A} = Q^{-1}AQ. \end{aligned}$$

Hence Q, the matrix of the eigenvectors of A, is the similarity transformation that takes A to a diagonal form.

Every matrix with all distinct eigenvalues is diagonalisable

Brief Review on Linear Algebra

Eigenvalues of A are not all distinct. An eigenvalue with multiplicity 2 or higher is called a *repeated* eigenvalue. An eigenvalue with multiplicity 1 is a *simple* eigenvalue.

When an eigenvalue appears repeated, say r times, it may not have r linearly independent eigenvectors. When there are less independent eigenvectors than eigenvalues, the matrix **cannot** have a diagonal representation.

Brief Review on Linear Algebra

Eigenvalues of A are not all distinct. An eigenvalue with multiplicity 2 or higher is called a *repeated* eigenvalue. An eigenvalue with multiplicity 1 is a *simple* eigenvalue.

When an eigenvalue appears repeated, say r times, it may not have r linearly independent eigenvectors. When there are less independent eigenvectors than eigenvalues, the matrix **cannot** have a diagonal representation.

An example of a non-diagonalisable matrix is

$$J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix},$$

which has the eigenvalue λ repeated 3 times, but only **one** independent eigenvector associated. The matrix J is a **Jordan block of order 3** associated with the eigenvalue λ.

Brief Review on Linear Algebra

For an eigenvalue λ repeated r times, there are $r+1$ possible Jordan block configurations. For example, for $r = 4$ we have

$\begin{bmatrix} \lambda & 1 & 0 & 0 \\ 0 & \lambda & 1 & 0 \\ 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$	one independent eigenvector	one Jordan block of order 4
$\begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 1 & 0 \\ 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$	two independent eigenvectors	one Jordan block of order 1, one Jordan block of order 3
$\begin{bmatrix} \lambda & 1 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$	two independent eigenvectors	two Jordan blocks of order 2
$\begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$	three independent eigenvectors	two Jordan blocks of order 1, ones Jordan block of order 2
$\begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix}$	four independent eigenvectors	four Jordan blocks of order 1

Brief Review on Linear Algebra

A matrix with repeated eigenvalues and a deficient number of associated eigenvectors cannot be diagonalised. However, it can always be taken to a **block-diagonal** and **triangular** form called the Jordan form.

Brief Review on Linear Algebra

A matrix with repeated eigenvalues and a deficient number of associated eigenvectors cannot be diagonalised. However, it can always be taken to a **block-diagonal** and **triangular** form called the Jordan form. For example,

$$\begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_2 \end{bmatrix}$$

This matrix has **two distinct eigenvalues**, λ_1 and λ_2; λ_1 is repeated five times, while λ_2 appears only once.

There are **two Jordan blocks** associated with λ_1; one of order 3 and one of order 2.

Brief Review on Linear Algebra

A matrix with repeated eigenvalues and a deficient number of associated eigenvectors cannot be diagonalised. However, it can always be taken to a **block-diagonal** and **triangular** form called the Jordan form. For example,

$$\begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_2 \end{bmatrix}$$

This matrix has **two distinct eigenvalues**, λ_1 and λ_2; λ_1 is repeated five times, while λ_2 appears only once.

There are **two Jordan blocks** associated with λ_1; one of order 3 and one of order 2.

For any square matrix $\mathbf{A}$, there is always a nonsingular matrix $\mathbf{Q}$ such that

$$\bar{\mathbf{A}} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}, \qquad \text{where } \bar{\mathbf{A}} \text{ is in \textbf{Jordan form}.}$$

Brief Review on Linear Algebra

Complex eigenvalues. The Jordan form applies also for a matrix with **complex eigenvalues**, but then it stops being a **real matrix**, e.g.,

$$\bar{A} = \begin{bmatrix} \sigma + j\omega & 1 & 0 & 0 \\ 0 & \sigma + j\omega & 0 & 0 \\ 0 & 0 & \sigma - j\omega & 1 \\ 0 & 0 & 0 & \sigma - j\omega \end{bmatrix}$$

Brief Review on Linear Algebra

Complex eigenvalues. The Jordan form applies also for a matrix with **complex eigenvalues**, but then it stops being a **real matrix**, e.g.,

$$\bar{A} = \begin{bmatrix} \sigma + j\omega & 1 & 0 & 0 \\ 0 & \sigma + j\omega & 0 & 0 \\ 0 & 0 & \sigma - j\omega & 1 \\ 0 & 0 & 0 & \sigma - j\omega \end{bmatrix}$$

Yet, it is still possible obtain a real matrix, the real Jordan form, which is still **block-diagonal**, although not anymore **triangular**.

$$\bar{A} = \begin{bmatrix} \sigma & \omega & 1 & 0 \\ -\omega & \sigma & 0 & 1 \\ 0 & 0 & \sigma & \omega \\ 0 & 0 & -\omega & \sigma \end{bmatrix} = \begin{bmatrix} B_{\sigma,\omega} & I \\ 0 & B_{\sigma,\omega} \end{bmatrix}$$

Brief Review on Linear Algebra

From the Jordan form of a matrix we can obtain important properties of its eigenvalues; two useful ones are

$$\boxed{\textbf{trace}\{A\} = \sum_{i=1}^{n} \lambda_i}, \qquad \boxed{\textbf{det}\{A\} = \prod_{i=1}^{n} \lambda_i}.$$

In MATLAB, `E=eig(A)` yields the vector `E` containing the eigenvalues of the square matrix `A`;

`[Q,D]=eig(A)` produces a diagonal matrix `D` of eigenvalues and a full matrix `Q` whose columns are the corresponding eigenvectors so that `A*Q = Q*E`.

`J=jordan(A)` computes the Jordan Canonical/Normal Form `J` of the matrix `A`. The matrix must be known exactly, so its elements must be integers or ratios of small integers.

Outline

- Brief Review on Linear Algebra
- Equivalent State Equations
- Canonical Forms
- Realisations

Equivalent State Equations

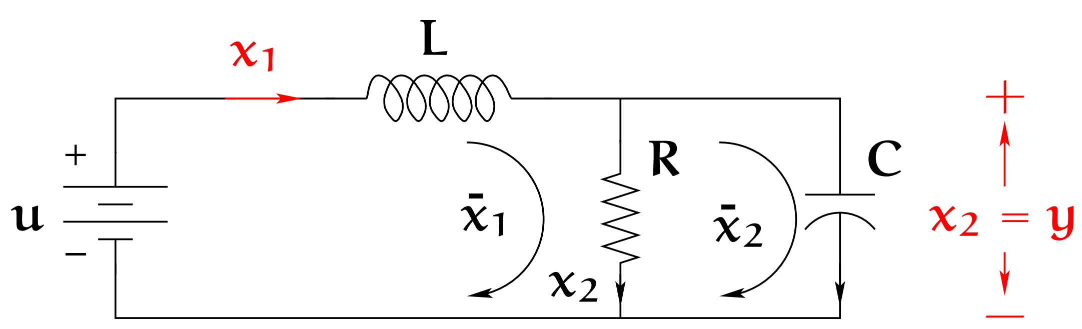
The state space description of a given system is **not unique**. Given a state space representation, a simple change of coordinates will take us to a different state space representation of the same system.

Equivalent State Equations

The state space description of a given system is **not unique**. Given a state space representation, a simple change of coordinates will take us to a different state space representation of the same system.

Example. Consider the RLC electric circuit of the figure where $R = 1\Omega$, $L = 1H$ and $C = 1F$. We take as output the voltage y across C.

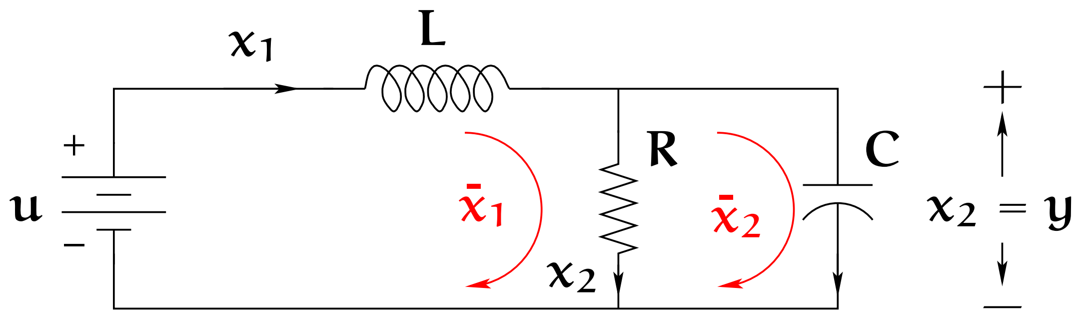
If we choose as state variables x_1, the current through the inductor L, and x_2, the voltage across the capacitor C, we get the state space description



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \qquad y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Equivalent State Equations

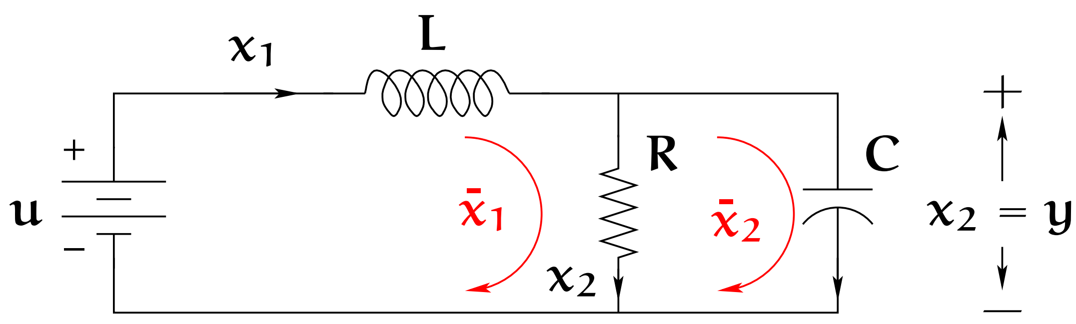
Example (continuation). On the other hand, if we choose as state variables the loop currents $\bar{x}_1$ and $\bar{x}_2$ we get the state space description



$$\begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$$

Equivalent State Equations

Example (continuation). On the other hand, if we choose as state variables the loop currents $\bar{x}_1$ and $\bar{x}_2$ we get the state space description



$$\begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$$

Both state equation descriptions represent the same circuit, so they must be closely related. In fact, we can verify that

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

i.e., $\bar{x} = \mathbf{P}x$ or $x = \mathbf{P}^{-1}\bar{x}$.

Equivalent State Equations

Algebraic Equivalence (AE): Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a nonsingular matrix, and let $\bar{x} = \mathbf{P}x$. Then the state equation

$$\begin{aligned} \dot{\bar{x}}(t) &= \bar{\mathbf{A}}\bar{x}(t) + \bar{\mathbf{B}}u(t) \\ y(t) &= \bar{\mathbf{C}}\bar{x}(t) + \bar{\mathbf{D}}u(t). \end{aligned} \quad \text{where} \quad \boxed{\bar{\mathbf{A}} = \mathbf{PAP}^{-1}}, \boxed{\bar{\mathbf{B}} = \mathbf{PB}}, \boxed{\bar{\mathbf{C}} = \mathbf{CP}^{-1}}, \boxed{\bar{\mathbf{D}} = \mathbf{D}},$$

is said to be *(algebraically) equivalent* to the state equation

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t). \end{aligned}$$

and $\bar{x} = \mathbf{P}x$ is called an *equivalence transformation*.

Equivalent State Equations

Algebraic Equivalence (AE): Let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, and let $\bar{\mathbf{x}} = \mathbf{P}\mathbf{x}$. Then the state equation

$$\begin{aligned}\dot{\bar{\mathbf{x}}}(t) &= \bar{\mathbf{A}}\bar{\mathbf{x}}(t) + \bar{\mathbf{B}}\mathbf{u}(t)\\ \mathbf{y}(t) &= \bar{\mathbf{C}}\bar{\mathbf{x}}(t) + \bar{\mathbf{D}}\mathbf{u}(t).\end{aligned} \quad \text{where} \quad \begin{array}{l}\boxed{\bar{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}}, \boxed{\bar{\mathbf{B}} = \mathbf{P}\mathbf{B}},\\ \boxed{\bar{\mathbf{C}} = \mathbf{C}\mathbf{P}^{-1}}, \boxed{\bar{\mathbf{D}} = \mathbf{D}},\end{array}$$

is said to be *(algebraically) equivalent* to the state equation

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t).\end{aligned}$$

and $\bar{\mathbf{x}} = \mathbf{P}\mathbf{x}$ is called an *equivalence transformation*.

From Linear Algebra, we know that the matrices $\mathbf{A}$ and $\bar{\mathbf{A}}$ are similar, and have the same eigenvalues. The MATLAB function `[Ab,Bb,Cb,Db] = ss2ss(A,B,C,D,P)` performs equivalence transformations between state space representations.

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$\bar{\mathbf{G}}(s) = \bar{\mathbf{C}}(s\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} + \bar{\mathbf{D}}$$

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$\begin{aligned}\bar{\mathbf{G}}(s) &= \bar{\mathbf{C}}(s\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} + \bar{\mathbf{D}} \\ &= \mathbf{C}\mathbf{P}^{-1}\left(s\mathbf{I} - \mathbf{P}\mathbf{A}\mathbf{P}^{-1}\right)^{-1}\mathbf{P}\mathbf{B} + \mathbf{D}\end{aligned}$$

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$
\begin{aligned}
\bar{\mathbf{G}}(s) &= \bar{\mathbf{C}}(s\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} + \bar{\mathbf{D}} \\
&= \mathbf{C}\mathbf{P}^{-1}\left(s\mathbf{I} - \mathbf{P}\mathbf{A}\mathbf{P}^{-1}\right)^{-1}\mathbf{P}\mathbf{B} + \mathbf{D} \\
&= \mathbf{C}\left(s\mathbf{P}^{-1}\mathbf{P} - \mathbf{P}^{-1}\mathbf{P}\mathbf{A}\mathbf{P}^{-1}\mathbf{P}\right)^{-1}\mathbf{B} + \mathbf{D}
\end{aligned}
$$

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$
\begin{aligned}
\bar{\mathbf{G}}(s) &= \bar{\mathbf{C}}(s\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} + \bar{\mathbf{D}} \\
&= \mathbf{C}\mathbf{P}^{-1}\left(s\mathbf{I} - \mathbf{P}\mathbf{A}\mathbf{P}^{-1}\right)^{-1}\mathbf{P}\mathbf{B} + \mathbf{D} \\
&= \mathbf{C}\left(s\mathbf{P}^{-1}\mathbf{P} - \mathbf{P}^{-1}\mathbf{P}\mathbf{A}\mathbf{P}^{-1}\mathbf{P}\right)^{-1}\mathbf{B} + \mathbf{D} \\
&= \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad = \mathbf{G}(s).
\end{aligned}
$$

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$\begin{aligned}\bar{G}(s) &= \bar{C}(sI - \bar{A})^{-1}\bar{B} + \bar{D} \\ &= CP^{-1}\left(sI - PAP^{-1}\right)^{-1}PB + D \\ &= C\left(sP^{-1}P - P^{-1}PAP^{-1}P\right)^{-1}B + D \\ &= C(sI - A)^{-1}B + D \quad = G(s).\end{aligned}$$

Sometimes, however, systems **not** necessarily AE may have the same transfer function.

Example. Consider the state equation

$$\begin{aligned}\dot{x}(t) &= -3x(t) + u(t) \\ y(t) &= 3x(t)\end{aligned}$$

Equivalent State Equations

Two AE (algebraically equivalent) state representations **have the same transfer function**, since

$$\begin{aligned}\bar{G}(s) &= \bar{C}(sI - \bar{A})^{-1}\bar{B} + \bar{D}\\ &= CP^{-1}\left(sI - PAP^{-1}\right)^{-1}PB + D\\ &= C\left(sP^{-1}P - P^{-1}PAP^{-1}P\right)^{-1}B + D\\ &= C(sI - A)^{-1}B + D \quad = G(s).\end{aligned}$$

Sometimes, however, systems **not** necessarily AE may have the same transfer function.

Example. Consider the state equation

$$\begin{aligned}\dot{x}(t) &= -3x(t) + u(t)\\ y(t) &= 3x(t)\end{aligned}$$

Its transfer function is $G(s) = \dfrac{3}{s+3}$.

Equivalent State Equations

Example (continuation). On the other hand, consider

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 3 & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$$

Equivalent State Equations

Example (continuation). On the other hand, consider

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 3 & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$$

Its transfer function is

$$G(s) = \begin{bmatrix} 3 & 0 \end{bmatrix} \times \begin{bmatrix} s+3 & 0 \\ 4 & s-1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \frac{1}{(s+3)(s-1)} \begin{bmatrix} 3 & 0 \end{bmatrix} \times \begin{bmatrix} s-1 & 0 \\ -4 & s+3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{3}{s+3}$$

The same as for the previous system, and they do not even have the same dimensions! □

Equivalent State Equations

We see that

$$\boxed{\text{Algebraic Equivalence}} \quad \begin{array}{c}\Rightarrow \\ \not\Leftarrow\end{array} \quad \boxed{\text{Same Transfer Function}}$$

Equivalent State Equations

We see that

Algebraic Equivalence	$\Rightarrow$ $\not\Leftarrow$	Same Transfer Function

A concept more general than that of AE is the following.

Zero-State Equivalence (ZSE): Two LTI state equations $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ and $\{\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}\}$ are zero-state equivalent if they have the same transfer (matrix) function.

Clearly, AE always implies ZSE, but the reverse does not hold.

The concepts of equivalence of state equations, AE and ZSE, are exactly the same for discrete-time LTI systems.

Outline

- Brief Review on Linear Algebra
- Equivalent State Equations
- Canonical Forms
- Realisations

Canonical Forms

Although for a system has an infinite number of state space representations, there are some particular forms of these state equations which present useful characteristics. These are known as **canonical forms**. We will discuss two of them:

- the Modal Canonical Form
- the Controller Canonical Form

Canonical Forms

Modal Canonical Form. A state equation in which the matrix $\mathbf{A}$ is in **Jordan form**. It is called *modal* because the eigenvalues (the *modes* of the system) are explicit in it.

To obtain the modal canonical form from an arbitrary state equation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ we have to use as **equivalence transformation** the matrix $\mathbf{P} = \mathbf{Q}^{-1}$, where $\mathbf{Q}$ is the similarity transformation that yields the Jordan form $\bar{\mathbf{A}}$ of the matrix $\mathbf{A}$.

Canonical Forms

Modal Canonical Form. A state equation in which the matrix $\mathbf{A}$ is in **Jordan form**. It is called *modal* because the eigenvalues (the *modes* of the system) are explicit in it.

To obtain the modal canonical form from an arbitrary state equation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ we have to use as **equivalence transformation** the matrix $\mathbf{P} = \mathbf{Q}^{-1}$, where $\mathbf{Q}$ is the similarity transformation that yields the Jordan form $\bar{\mathbf{A}}$ of the matrix $\mathbf{A}$.

Example. Consider state equation $\dot{x} = \mathbf{A}x + \mathbf{B}u$, $y = \mathbf{C}x + \mathbf{D}u$, with

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 2 \\ -2 & 2 & 2 \\ 0 & -1 & 2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The eigenvalues of $\mathbf{A}$ are $\lambda_1 = 1 + j$, $\lambda_2 = 1 - j$, and $\lambda_3 = 2$, respectively with eigenvectors

$$v_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - j \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + j \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Canonical Forms

Example (continuation). The equivalence transformation $\mathbf{Q} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ takes $\mathbf{A}$ to the real Jordan form

$$\bar{\mathbf{A}} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The transformed matrices $\bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}$ are

$$\bar{\mathbf{B}} = \mathbf{Q}^{-1}\mathbf{B} = \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix}, \quad \bar{\mathbf{C}} = \mathbf{C}\mathbf{Q} = \begin{bmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \end{bmatrix}, \quad \bar{\mathbf{D}} = \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The state equation given by $\{\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}\}$ is in **modal canonical form**. □

Canonical Forms

Controller Canonical Form. A state equation in which the matrix A is in **companion form** with the coefficients of its characteristic polynomial on the first row.

This canonical form will be useful to explain state feedback control design. In the SISO case the matrices have the form

$$\bar{A} = \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} \beta_1 & \beta_2 & \ldots & \beta_{n-1} & \beta_n \end{bmatrix}, \qquad \bar{D} = \gamma.$$

The matrices $\bar{C}$ and $\bar{D}$ have no special structure.

Canonical Forms

Let $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ be a generic SISO state equation representation (of order 4, for simplicity), in which the **characteristic polynomial** of $\mathbf{A}$ is $\Delta(\lambda) = \lambda^4 + \alpha_1\lambda^3 + \alpha_2\lambda_2^2 + \alpha_3\lambda + \alpha_4$. To obtain the **Controller Canonical Form** of this system we introduce the matrices

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Canonical Forms

Let $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ be a generic SISO state equation representation (of order 4, for simplicity), in which the **characteristic polynomial** of $\mathbf{A}$ is $\Delta(\lambda) = \lambda^4 + \alpha_1\lambda^3 + \alpha_2\lambda_2^2 + \alpha_3\lambda + \alpha_4$. To obtain the **Controller Canonical Form** of this system we introduce the matrices

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & 1 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, under the assumption that $\mathcal{C}$ is **nonsingular** the equivalence transformation

$$\mathbf{P} = (\mathcal{C}\mathbf{R})^{-1}$$

yields the matrices $\bar{\mathbf{A}} = \mathbf{PAP}^{-1}$, $\bar{\mathbf{B}} = \mathbf{PB}$, $\bar{\mathbf{C}} = \mathbf{CP}^{-1}$, $\bar{\mathbf{D}} = \mathbf{D}$ in Controller Canonical Form.

The matrix $\mathcal{C}$ is called the **Controllability Matrix**.

Canonical Forms

The Controller Canonical Form provides a **direct method of obtaining a state equation from a transfer matrix** (a realisation).

Indeed, it is not difficult to check that

$$\bar{\mathbf{A}} = \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\bar{\mathbf{C}} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_{n-1} & \beta_n \end{bmatrix}, \qquad \bar{\mathbf{D}} = \gamma$$

yields the transfer function

$$\mathbf{G}(s) = \frac{\beta_1 s^{n-1} + \beta_2 s^{n-2} + \cdots + \beta_n}{s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_n} + \gamma.$$

Canonical Forms

The Controller Canonical Form provides a **direct method of obtaining a state equation from a transfer matrix** (a realisation).

Indeed, it is not difficult to check that

$$\bar{A} = \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} \beta_1 & \beta_2 & \ldots & \beta_{n-1} & \beta_n \end{bmatrix}, \qquad \bar{D} = \gamma$$

yields the transfer function

$$G(s) = \frac{\beta_1 s^{n-1} + \beta_2 s^{n-2} + \cdots + \beta_n}{s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_n} + \gamma.$$

Hence, for a given transfer function $G(s)$, we can directly obtain a state equation representation from the coefficients of its numerator, denominator, and high frequency gain.

Canonical Forms

Given a SISO state equation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, the following MATLAB code computes its Controller Canonical Form

```
G = ss(A,B,C,D); % system in original coordinates
pol=poly(G.a); % get characteristic polynomial
n=length(G.a); % get system order
CC=ctrb(G.a,G.b);% get controllability matrix
R=toeplitz(eye(n,1),pol(1:n-1)); % built R
P=inv(CC*R); % built equiv. transformation P
Gbar=ss2ss(G,P); % transform to CCF
```

Neither the Controller Canonical Form or the Modal Canonical Form are recommended for numerical computations for large order systems, since they are generally ill-conditioned.

Nevertheless, these canonical forms have great value to **analyse** and **understand** state equation system theory.

Outline

- Brief Review on Linear Algebra
- Equivalent State Equations
- Canonical Forms
- Realisations

Realisations

- So far, we know that a LTI system can be represented by the **external description** given by its transfer (matrix) function

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s).$$

Realisations

- So far, we know that a LTI system can be represented by the **external description** given by its transfer (matrix) function

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s).$$

- If the system is also finite dimensional (lumped) it can also be represented by the **internal description** given by state equations

$$\begin{aligned}\dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t)\\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t).\end{aligned}$$

Realisations

- So far, we know that a LTI system can be represented by the **external description** given by its transfer (matrix) function

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s).$$

- If the system is also finite dimensional (lumped) it can also be represented by the **internal description** given by state equations

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t).\end{aligned}$$

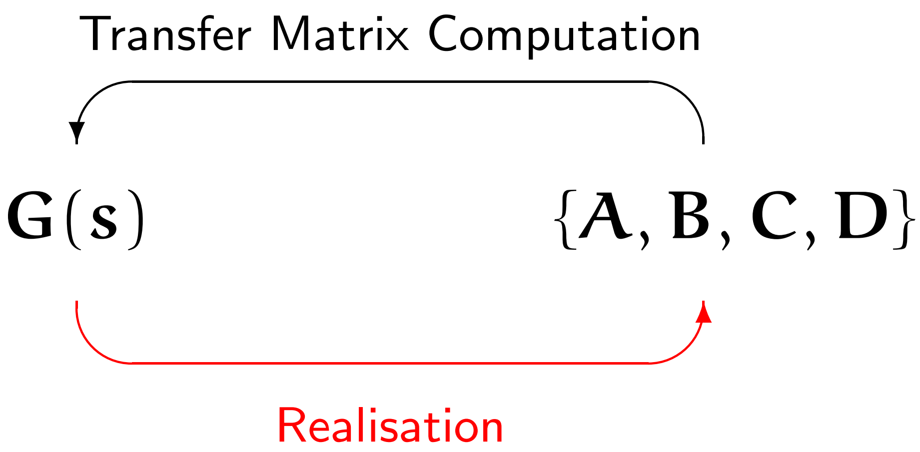
- If the state equations of the system are known, then the transfer matrix can be computed from the system matrices as

$$G(s) = C(sI - A)^{-1}B + D,$$

and this computed transfer matrix is **unique**.

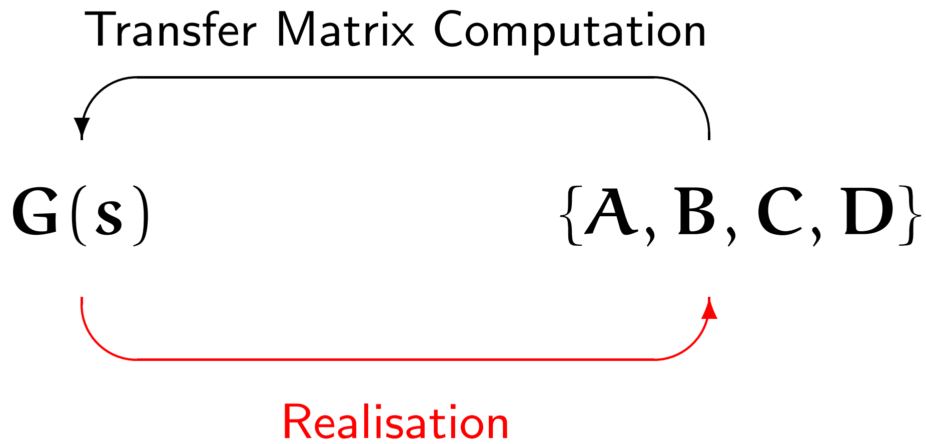
Realisations

The realisation problem is the converse to obtaining $\mathbf{G}(s)$ from $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$. That is, it is the problem of obtaining the system state equations from its transfer matrix.



Realisations

The realisation problem is the converse to obtaining $\mathbf{G}(s)$ from $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$. That is, it is the problem of obtaining the system state equations from its transfer matrix.



A transfer matrix $\mathbf{G}(s)$ is said to be realisable if there exists a finite-dimensional state equation, or simply a quadruple $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ such that

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}.$$

The quadruple $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ is then called a realisation of $\mathbf{G}(s)$.

Realisations

- Although for a given quadruple $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ the transfer matrix $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ is unique, a given transfer matrix $\mathbf{G}(s)$ does **not** have a unique realisation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$.

Realisations

- Although for a given quadruple $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ the transfer matrix $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ is unique, a given transfer matrix $\mathbf{G}(s)$ does **not** have a unique realisation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$.
- Different realisations present different properties (e.g., good numerical condition, minimal order, etc.) which might be convenient depending on their application.

Realisations

- Although for a given quadruple $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ the transfer matrix $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ is unique, a given transfer matrix $\mathbf{G}(s)$ does **not** have a unique realisation $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$.

- Different realisations present different properties (e.g., good numerical condition, minimal order, etc.) which might be convenient depending on their application.

Theorem (Realisability). A transfer matrix $\mathbf{G}(s)$ is realisable if and only if $\mathbf{G}(s)$ is a proper rational transfer matrix.

Recall that a rational (i.e., quotient of polynomials) transfer function is **proper** if the degree of its numerator is not greater than that of its denominator. A transfer matrix is proper if all its elements are proper transfer functions.

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:
  - $\mathbf{A}$ is obtained from the coefficients of the characteristic polynomial, $\alpha_1, \alpha_2, \ldots, \alpha_n$, from the denominator of $\mathbf{G}(s)$

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:
  - $\mathbf{A}$ is obtained from the coefficients of the characteristic polynomial, $\alpha_1, \alpha_2, \ldots, \alpha_n$, from the denominator of $\mathbf{G}(s)$
  - $\mathbf{B}$ is trivial

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:
  - $\mathbf{A}$ is obtained from the coefficients of the characteristic polynomial, $\alpha_1, \alpha_2, \ldots, \alpha_n$, from the denominator of $\mathbf{G}(s)$
  - $\mathbf{B}$ is trivial
  - $\mathbf{D} = \lim_{s\to\infty} \mathbf{G}(s)$, the direct feedthrough, a.k.a. high-frequency gain; needs to be obtained before $\mathbf{C}$

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:
  - $\mathbf{A}$ is obtained from the coefficients of the characteristic polynomial, $\alpha_1, \alpha_2, \ldots, \alpha_n$, from the denominator of $\mathbf{G}(s)$
  - $\mathbf{B}$ is trivial
  - $\mathbf{D} = \lim_{s\to\infty} \mathbf{G}(s)$, the direct feedthrough, a.k.a. high-frequency gain; needs to be obtained before $\mathbf{C}$
  - $\mathbf{C}$ is obtained from the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ of the numerator of $\mathbf{G}(s) - \mathbf{D}$.

Realisations

- For a SISO system we know a direct method to obtain a state equation in the Controller Canonical Form:
  - $\mathbf{A}$ is obtained from the coefficients of the characteristic polynomial, $\alpha_1, \alpha_2, \ldots, \alpha_n$, from the denominator of $\mathbf{G}(s)$
  - $\mathbf{B}$ is trivial
  - $\mathbf{D} = \lim_{s\to\infty} \mathbf{G}(s)$, the direct feedthrough, a.k.a. high-frequency gain; needs to be obtained before $\mathbf{C}$
  - $\mathbf{C}$ is obtained from the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ of the numerator of $\mathbf{G}(s) - \mathbf{D}$.
- For a SIMO system, say p outputs, we can use the same direct method; the only alterations are in $\mathbf{C}$ and $\mathbf{D}$,

$$\mathbf{D} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_p \end{bmatrix} = \lim_{s\to\infty} \begin{bmatrix} G_1(s) \\ G_2(s) \\ \vdots \\ G_p(s) \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \ldots & \beta_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ \beta_{p1} & \beta_{p2} & \ldots & \beta_{pn} \end{bmatrix}$$

Realisations

- For a MIMO system , say p outputs and q inputs, we can still use the direct method, by considering the system as **the superposition of several SIMO systems**,

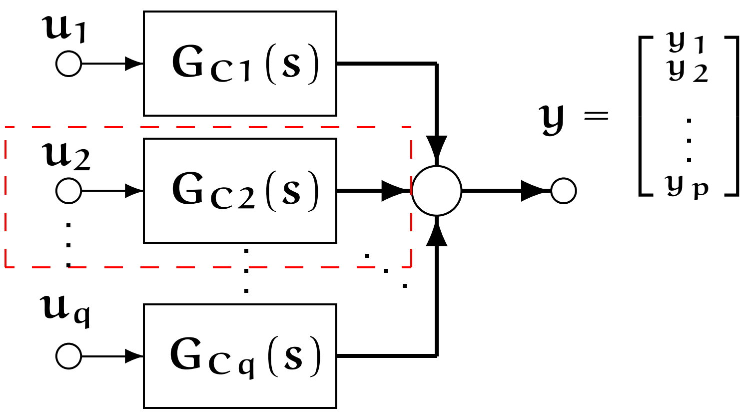
$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ \vdots \\ y_p(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & \cdots & \vdots \\ G_{p1}(s) & G_{p2}(s) & \cdots & G_{pq}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_q(s) \end{bmatrix}$$

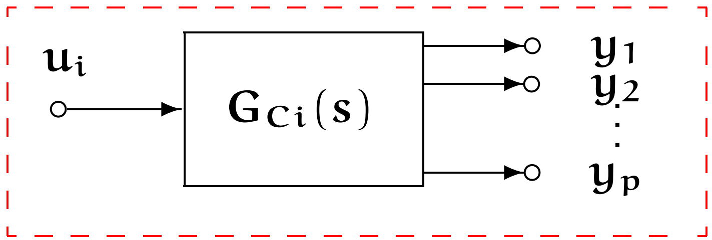
$$= \begin{bmatrix} G_{C1}(s) & G_{C2}(s) & \cdots & G_{Cq}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ \vdots \\ u_p(s) \end{bmatrix}$$

$$= G_{C1}(s)u_1(s) + G_{C2}(s)u_2(s) + \cdots + G_{Cq}(s)u_q$$

Realisations

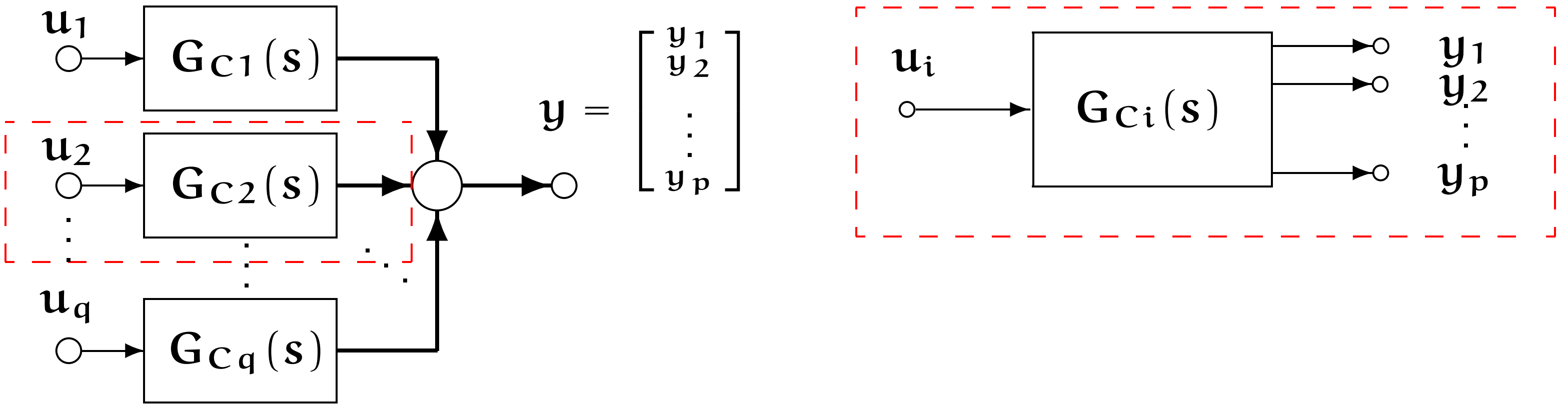
A MIMO system as the superposition of several SIMO systems:





Realisations

A MIMO system as the superposition of several SIMO systems:



If A_i, B_i, C_i, D_i is the realisation of column $G_{Ci}(s)$, $i = 1, \dots, m$, of $G(s)$, then a realisation of the superposition is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_q \end{bmatrix} = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix} + \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_q \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_q \end{bmatrix}$$

$$y = \begin{bmatrix} C_1 & C_2 & \cdots & C_q \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix} + \begin{bmatrix} D_1 & D_2 & \cdots & D_q \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_q \end{bmatrix}$$

Realisations

Example. Consider the 2×2 transfer matrix

$$\mathbf{G}(s) = \begin{bmatrix} \frac{4s-10}{2s+1} & \frac{3}{s+2} \\ \frac{1}{(2s+1)(s+2)} & \frac{s+1}{(s+2)^2} \end{bmatrix}.$$

We first separate the direct gain $\mathbf{D}$ and the strictly proper part $\check{\mathbf{G}}(s)$

$$\mathbf{G}(s) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \frac{-12}{2s+1} & \frac{3}{s+2} \\ \frac{1}{(2s+1)(s+2)} & \frac{s+1}{(s+2)^2} \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{G}(\infty)=\mathbf{D}} + \underbrace{\begin{bmatrix} \frac{\begin{bmatrix} -6(s+2) \\ 1/2 \end{bmatrix}}{s^2+\frac{5}{2}s+1} & \frac{\begin{bmatrix} 3(s+2) \\ (s+1) \end{bmatrix}}{(s+2)^2} \end{bmatrix}}_{\check{\mathbf{G}}(s) \text{ strictly proper part}}$$

Note per-column common denominator

Realisations

Example (continuation). We realise the strictly proper part $\check{G}(s)$ by columns. A realisation for the first column of $\check{G}(s)$ is

$$\left[\frac{\begin{bmatrix} -6(s+2) \\ 1/2 \end{bmatrix}}{s^2+\frac{5}{2}s+1}\right] \quad \rightleftarrows \quad \begin{aligned} \dot{x}_1 &= \begin{bmatrix} -\frac{5}{2} & -1 \\ 1 & 0 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 \\ y_{C1} &= \begin{bmatrix} -6 & -12 \\ 0 & \frac{1}{2} \end{bmatrix} x_1 \end{aligned}$$

And a realisation for the second column of $\check{G}(s)$ is

$$\left[\frac{\begin{bmatrix} 3(s+2) \\ (s+1) \end{bmatrix}}{s^2+4s+4}\right] \quad \rightleftarrows \quad \begin{aligned} \dot{x}_2 &= \begin{bmatrix} -4 & -4 \\ 1 & 0 \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_2 \\ y_{C1} &= \begin{bmatrix} 3 & 6 \\ 1 & 1 \end{bmatrix} x_2 \end{aligned}$$

Finally, we superpose the column realisations to get that of $G(s)$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{5}{2} & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -4 & -4 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -6 & -12 & 3 & 6 \\ 0 & \frac{1}{2} & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Realisations

We summarise the procedure used in the example as it is useful to find a realisation of any (even non-square) transfer matrix.

CCF Realisation Procedure. Start with a given transfer matrix $\mathbf{G}(s)$

1. Compute the **high-frequency gain** matrix $\mathbf{D} = \lim_{s\to\infty} \mathbf{G}(s)$.
2. Obtain the **strictly proper part** of $\mathbf{G}(s)$ i.e., $\check{\mathbf{G}}(s) = \mathbf{G}(s) - \mathbf{D}$.
3. If the system has more than one input ($\mathbf{G}(s)$ is $p \times q$, with $q > 1$) split $\check{\mathbf{G}}(s)$ in columns $\check{\mathbf{G}} = [\, \check{G}_{C1} \; \check{G}_{C2} \; \dots \; \check{G}_{Cq} \,]$, obtaining per-column common denominators.
4. Obtain a CCF realisation $\{A_i, B_i, C_i\}$ of each $\check{\mathbf{G}}_{Ci}$ for $i = 1 : q$.
5. Form the realisation of $\mathbf{G}(s)$ as

$$A = \textbf{blockdiag}[A_1, A_2, \dots, A_q], \quad C = [\, C_1 \; C_2 \; \dots \; C_q \,],$$
$$B = \textbf{blockdiag}[B_1, B_2, \dots, B_q], \quad D$$

Realisations

- Notice that this direct method to obtain a state equation realisation of a transfer matrix does not necessarily give a realisation with as many eigenvalues of A as poles in $\mathbf{G}(s)$.

  Generally, we will obtain **more** eigenvalues than poles in $\mathbf{G}(s)$.

Realisations

- Notice that this direct method to obtain a state equation realisation of a transfer matrix does not necessarily give a realisation with as many eigenvalues of A as poles in $\mathbf{G}(s)$.

  Generally, we will obtain **more** eigenvalues than poles in $\mathbf{G}(s)$.

- For any given transfer matrix $\mathbf{G}(s)$ there always exist realisations of **minimal order**, in which, if $\mathbf{G}(s)$ has n poles, say, the matrix A is the realisation is $n \times n$, i.e., it has n eigenvalues. These realisations are called **minimal**.

Realisations

- Notice that this direct method to obtain a state equation realisation of a transfer matrix does not necessarily give a realisation with as many eigenvalues of A as poles in $G(s)$.

  Generally, we will obtain **more** eigenvalues than poles in $G(s)$.

- For any given transfer matrix $G(s)$ there always exist realisations of **minimal order**, in which, if $G(s)$ has n poles, say, the matrix A is the realisation is $n \times n$, i.e., it has n eigenvalues. These realisations are called **minimal**.

- A nonminimal realisation can still produce the same transfer function $G(s)$ because there will be pole-zero cancellations in $C(sI - A)^{-1}B + D$ that make the "excess" eigenvalues disappear in the resulting transfer matrix.

Realisations

The MATLAB function to obtain a minimal realisation is `Gmr=minreal(G)`, or `[Am,Bm,Cm,Dm]=minreal(A,B,C,D)`. For the example, the following MATLAB code

```
A=[-5/2,-1 0 0;1 0 0 0;0 0 -4 -4;0 0 1 0];
B=[1 0;0 0;0 1;0 0];
C=[-6 -12 3 6;0 1/2 1 1];
D=[2 0;0 0];

G=ss(A,B,C,D);
Gmr=minreal(G);
```

yields the minimal realisation

$$\mathbf{A} = \begin{bmatrix} -0.4198 & -0.3802 & -0.3654 \\ 0.642 & -3.842 & -3.523 \\ -0.321 & 0.921 & -0.2383 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.4 & 0.08889 \\ -0.4 & 0.9111 \\ 0.2 & 0.04444 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} -13.33 & 4.333 & 5.333 \\ 0.5 & 1 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Realisations

A **minimal realisation** is intrinsically related to the **controllability** and **observability** properties of a state equation, as we will see later.

Realisations in Discrete-Time Systems

Discrete-time state equations. The realisation issues for discrete-time state equations are exactly the same as for continuous-time state equations, since the relation between state matrices and transfer function is the same,

$$\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

$$\Updownarrow$$

$$\begin{aligned} x[k+1] &= \mathbf{A}x[k] + \mathbf{B}u[k] \\ y[k] &= \mathbf{C}x[k] + \mathbf{D}u[k] \end{aligned}$$

Summary

- We reviewed some basic concepts of Linear Algebra required for the course: eigenvalues and eigenvectors, diagonal and Jordan form, etc.

Summary

- We reviewed some basic concepts of Linear Algebra required for the course: eigenvalues and eigenvectors, diagonal and Jordan form, etc.
- We presented the concept of **algebraic equivalence** and **zero state equivalence** between state equations.

Summary

- We reviewed some basic concepts of Linear Algebra required for the course: eigenvalues and eigenvectors, diagonal and Jordan form, etc.
- We presented the concept of **algebraic equivalence** and **zero state equivalence** between state equations.
- We studied two important **canonical forms** of state equations: the Modal Canonical Form and the Controller Canonical Form, which will be used in future lectures.

Summary

- We reviewed some basic concepts of Linear Algebra required for the course: eigenvalues and eigenvectors, diagonal and Jordan form, etc.
- We presented the concept of **algebraic equivalence** and **zero state equivalence** between state equations.
- We studied two important **canonical forms** of state equations: the Modal Canonical Form and the Controller Canonical Form, which will be used in future lectures.
- We discussed the problem of **realisation** of a transfer matrix, and presented a (*not necessarily minimal*) procedure to obtain a realisation of an arbitrary proper transfer matrix $\mathbf{G}(s)$ using the CCF.