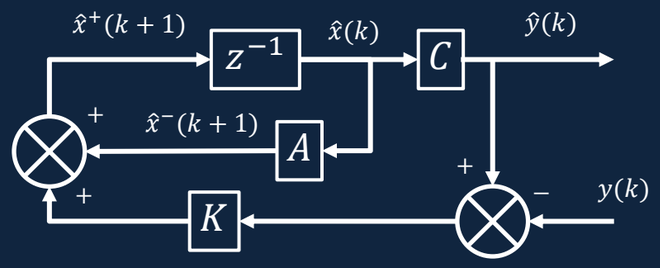


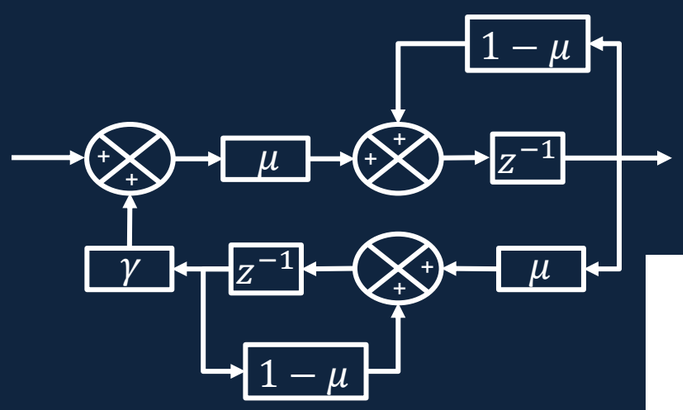
Discover the future of systems control with this innovative book on identification and adaptive control.

Written by an industrial engineer with specialized training in automatic control and machine learning, this book offers a modern and practical perspective on the design of adaptive systems applied to real-world environments, guiding readers step by step from linear control theory to nonlinear adaptive control.

It delves deeply into the theoretical and practical foundations of systems control, with a focus on online machine learning. Ideal for students, researchers, and engineers, it presents powerful tools for addressing the identification and control of nonlinear systems in real time.

The book includes practical examples programmed in MATLAB, freely available on the book's website, enabling applied understanding of each concept. Additionally, it contains detailed mathematical demonstrations to strengthen comprehension of the fundamentals of adaptive control.

Whether you are beginning in the field or looking to expand your skills in intelligent control, this book is an essential guide to the control technologies of the future.



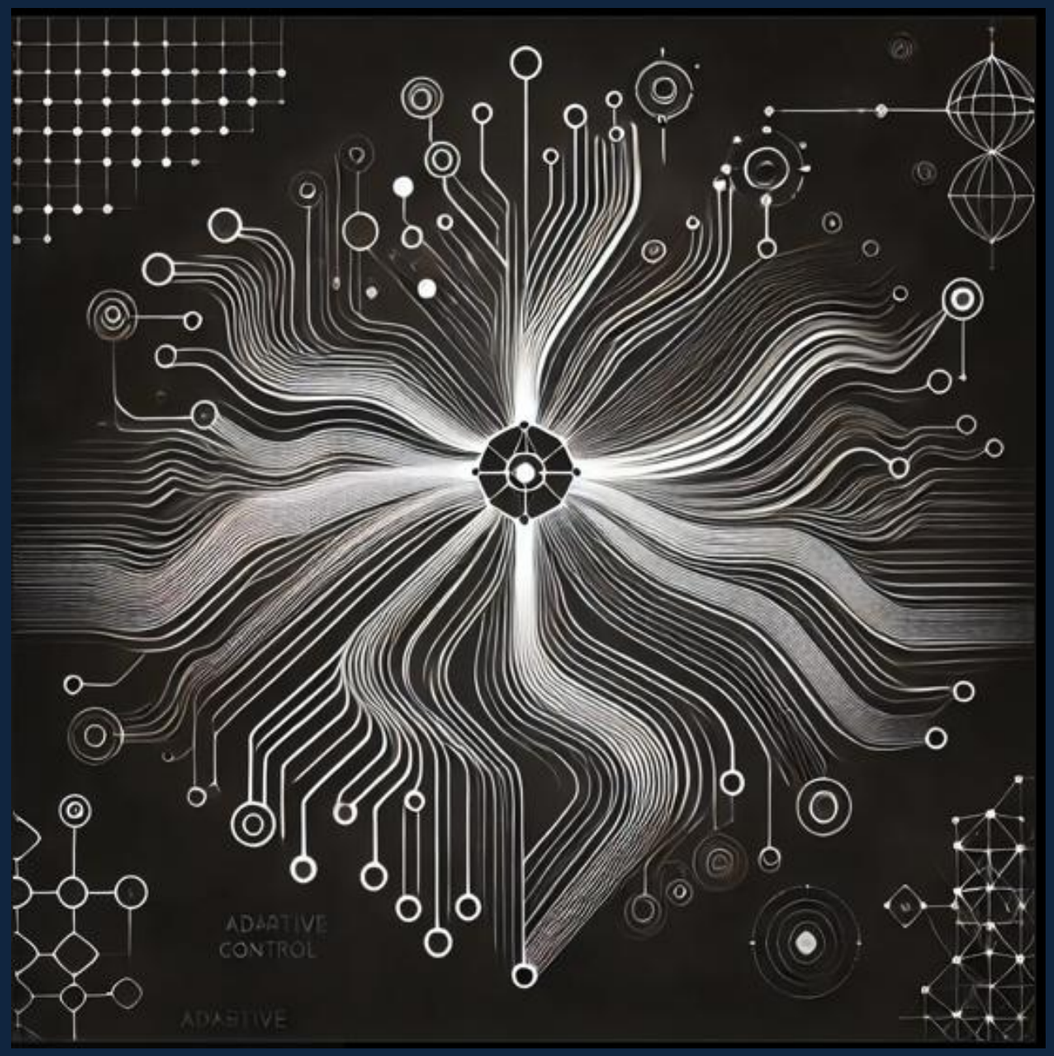
9788409746514

NEURAL NETWORKS AND ADAPTIVE CONTROL

C.A.L SEGURA

Neural Networks and Adaptive Control

AN ONLINE MACHINE LEARNING PERSPECTIVE



C.A.L SEGURA

NEURAL NETWORKS AND ADAPTIVE CONTROL

AN ONLINE MACHINE LEARNING PERSPECTIVE

CAL Segura

2025

ISBN (Hardcover): 978-84-09-77486-9

Published by the author/editor
CAL Segura
neuralnetadaptivecontrol@gmail.com

Title: Neural Networks and Adaptive Control
Subtitle: An Online Machine Learning Perspective
Author: CAL Segura
Edition: 1st Special Edition (Translation from Spanish)
Entry date (Spanish edition): September 7, 2025
Spanish Copyright Registration: Nº expediente 00765-02815209
Registry: Registro Central de la Propiedad Intelectual, Spain

English version of the work originally published in Spanish:
Title: Neural Networks and Adaptive Control
Subtitle: Una perspectiva del aprendizaje automático en línea
Author: CAL Segura

Format: Hardcover (U.S.)

IBIC / THEMA Identifiers:
TJFM1 – Robotics (Featured Topic)
4T – For specific educational purposes
UP – Practical applications of information technology
GPFC – Cybernetics and systems theory
TJFM – Automatic control engineering

Acknowledgments

I deeply thank my daughter for her love and my partner for her patience and constant support.

About the Author

C.A.L Segura holds degrees in Industrial Engineering and Electronic Engineering, with a specialization in control systems and automation. After extensive professional experience, he decided to pursue postgraduate studies at a prestigious international institution, where he delved into research areas related to adaptive control and machine learning. He is currently engaged in independent technical publishing, focusing his work on the intersection of artificial intelligence and control theory.

Website

https://www.neuralnetadaptivecontrol.com/
neuralnetadaptivecontrol@gmail.com

The official website of the book serves as a complementary digital platform, designed to expand and keep up to date the contents presented in this work. It is aimed at students, researchers, and professionals interested in adaptive control, state-space control, as well as its intersection with machine learning, neural networks, and convex optimization.

Contact the author at neuralnetadaptivecontrol@gmail.com

Notation

This section explains in detail the notation that will be used throughout this document.

States and Signals

$x(t)$	State vector in continuous time
$\dot{x}(t)$	Time derivative of the continuous-time state
$x(k)$	State vector in discrete time
$x(k+1)$	State at the next discrete instant
$u(t)$	System input in continuous time
$u(k)$	System input in discrete time
$y(t)$	System output in continuous time
$y(k)$	System output in discrete time
T_s	Sampling period

System Matrices

A_c, B_c, C_c, D_c	Continuous state-space system matrices
A_d, B_d, C_d, D_d	Discrete state-space system matrices
I	Identity matrix of implicit dimension

Laplace Transform and Transfer Functions

s	Complex variable used in the Laplace transform
$\mathcal{L}\{\cdot\}$	Laplace transform
$\mathcal{L}^{-1}\{\cdot\}$	Inverse Laplace transform
$X(s), Y(s), U(s)$	Laplace transforms of $x(t), y(t), u(t)$
$\Phi(s)$	State transition function: $C_c(sI - A_c)^{-1}$
$\Upsilon(s)$	Generalized transfer function: $C_c(sI - A_c)^{-1}B_c + D_c$
$H(s), H_{yu}(s)$	Continuous transfer functions

Z Transform and Transfer Functions

z	Complex variable used in the Z transform
$Z\{\cdot\}$	Z transform
$\mathcal{Z}^{-1}\{\cdot\}$	Inverse Z transform
$X(z), Y(z), U(z)$	Z transforms of $x(k), y(k), u(k)$
$\Phi(z)$	State transition function: $C_d(zI - A_d)^{-1}$
$\Upsilon(z)$	Generalized transfer function: $C_d(zI - A_d)^{-1}B_d + D_d$
$H(z), H_{yu}(z)$	Discrete transfer functions

Modeling and Learning

$e[k]$	Error between actual and estimated output
$\hat{y}[k]$	Estimated system output
$\varepsilon[k]$	Model error
$J[k]$	Instantaneous cost: $\frac{1}{2}e[k]^2$
η	Learning rate
$\boldsymbol{\phi}[k]$	Feature vector
$\mathbf{w}[k]$	Adaptive parameter vector
$\Delta\mathbf{w}[k]$	Change in adaptive parameters
$\mathbf{n}$	Vector of past inputs
$\hat{\mathbf{y}}$	Vector of past estimated outputs
$\mathbf{X}_k$	Combined vector for inner product
$\nabla_\theta \mathcal{L}$	Gradient of the loss function with respect to parameters θ

Neural Networks and Functions

$f(x)$	Target function to approximate
$g(x)$	Neural network approximation
ϵ	Approximation tolerance
$\mathbf{W}^{(\ell)}$	Weight matrix of layer ℓ
$\mathbf{b}^{(\ell)}$	Bias vector of layer ℓ
$\mathbf{a}^{(\ell)}$	Activations of layer ℓ
$\mathbf{z}^{(\ell)}$	Weighted input to layer ℓ: $\mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{b}^{(\ell)}$
$\delta^{(\ell)}$	Back-propagated error (delta) in layer ℓ
L	Total number of layers (includes hidden and output layers)

Linear Algebra and Operations

$\text{adj}(\boldsymbol{A})$	Adjugate matrix of $\boldsymbol{A}$
$\det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	Inner product between vectors
$\|\boldsymbol{x}\|$	Norm of vector $\boldsymbol{x}$
$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\odot$	Element-wise product (Hadamard)
$\otimes$	Outer product between vectors or matrices
$\circ$	Composition of functions
∇	Gradient operator

Contents

Chapter 1

Linear Algebra Tools

> "The century of data has begun! The truth is that vectors and matrices have become the language to know." Gilbert Strang Strang [2016]

This chapter introduces basic tools of linear algebra that will be used throughout the mathematical developments in this book. The automatic control of discrete systems, system modeling, and, more generally, the online study of linear and nonlinear systems rely on the acquisition or estimation of numerical values in real time. The historical numerical data generated by any industrial process, captured through data acquisition systems, have a direct mathematical correspondence with vectors and matrices in an n-dimensional space.

Consequently, the processing of a single time series—or a set of them—is directly connected to the fundamental properties of linear algebra. Furthermore, in the context of adaptive systems or systems employing online convex optimization, the analysis of these time series allows us to uncover the essential characteristics of the system or process under study.

For these reasons, this chapter presents several linear algebra tools that will be developed and applied throughout the book. These tools provide a formal framework to link historical sensor data—essentially time series represented as vectors or matrices—with the underlying algebraic properties that govern their behavior.

1.1 Dot Product or Scalar Product

The dot product, also known as the scalar product and called *dot product* in English, is an operation between two vectors of the same size that results in a real number (scalar). It is used to measure the angular relationship between vectors and has fundamental applications in linear algebra, geometry, physics, and machine learning.

Algebraic Definition

Given two vectors in $\mathbb{R}^n$:

$$\mathbf{u} = (u_1, u_2, \ldots, u_n), \quad \mathbf{v} = (v_1, v_2, \ldots, v_n) \tag{1.1}$$

The dot product is defined as:

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n = \sum_{i=1}^{n} u_i v_i \tag{1.2}$$

Geometric Definition

It can also be expressed in terms of the angle θ between the vectors:

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \cdot \|\mathbf{v}\| \cdot \cos(\theta) \tag{1.3}$$

where $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ represent the norms (lengths) of the vectors, and θ is the angle between them.

If $\mathbf{u} \neq \mathbf{0}$ and $\mathbf{v} \neq \mathbf{0}$, the cosine of the angle can be solved as:

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \tag{1.4}$$

This expression is useful to determine whether two vectors are orthogonal $\cos(\theta) = 0$, parallel $\cos(\theta) = \pm 1$, or form any other angle between them.

Properties

- Scalar compatibility: $(c\,\mathbf{u}) \cdot \mathbf{v} = c\,(\mathbf{u} \cdot \mathbf{v})$
- Orthogonality: If $\mathbf{u} \cdot \mathbf{v} = 0$, then the vectors are orthogonal (perpendicular).

Cauchy–Schwarz Inequality

For any pair of vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$|\mathbf{u} \cdot \mathbf{v}| \leq \|\mathbf{u}\| \cdot \|\mathbf{v}\| \tag{1.5}$$

Equality holds if and only if $\mathbf{u}$ and $\mathbf{v}$ are linearly dependent.

Triangle Inequality

For any pair of vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\| \tag{1.6}$$

Equality holds if $\mathbf{u}$ and $\mathbf{v}$ are collinear and point in the same direction.

Inner and Outer Product

- **Inner product**: Given two column vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, their inner product is

$$\mathbf{u}^\top \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{n} u_i v_i, \tag{1.7}$$

which results in a scalar.

- **Outer product**: Given $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$, the outer product is an $m \times n$ matrix defined by

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top, \tag{1.8}$$

where each element is $(\mathbf{u}\mathbf{v}^\top)_{ij} = u_i v_j$.

1.2 Norms and Lengths

Typically, we use the notation $\|\mathbf{A}\|$ to refer to the norm of a matrix and $\|\mathbf{v}\|$ to indicate the length or magnitude of a vector. However, it is common in mathematical and scientific literature for both terms, norm and length, to be used interchangeably, especially in the context of vectors.

Norms provide a way to quantify the size or magnitude of a vector or matrix according to different criteria. In this section, we present the most commonly used norms: the **Euclidean norm**, also called the 2-norm, its squared version, and the generalized *p-norm*.

2-Norm or Vector Length

The **2-norm**, also known as the **Euclidean length**, measures the distance from the origin to the point defined by the vector $\mathbf{v} \in \mathbb{R}^n$:

$$\|\mathbf{v}\| = \|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} = \sqrt{\mathbf{v} \cdot \mathbf{v}} = (\mathbf{v} \cdot \mathbf{v})^{1/2} \tag{1.9}$$

It is the most widely used norm in geometry, physics, and engineering.

Squared Length or Squared 2-Norm

The **squared length** of a vector is simply the square of its 2-norm. It is defined as:

$$\|\mathbf{v}\|^2 = \|\mathbf{v}\|_2^2 = \mathbf{v} \cdot \mathbf{v} = v_1^2 + v_2^2 + \cdots + v_n^2 \tag{1.10}$$

This form is frequently used in computational and optimization contexts, as it avoids calculating the square root.

p-Norm

The p**-norm** of a vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$ is defined for $p \geq 1$ as:

$$\|\mathbf{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p} \tag{1.11}$$

Important special cases are:

- $\|\mathbf{v}\|_1 = \sum_{i=1}^{n} |v_i|$: Manhattan norm or ℓ_1
- $\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$: Euclidean norm or ℓ_2
- $\|\mathbf{v}\|_\infty = \max_i |v_i|$: infinity norm

All these norms satisfy the fundamental properties: positivity, homogeneity, zero-vector identity, and the triangle inequality.

Unit Vector

A *unit vector* is a vector whose norm is equal to one. It is obtained by normalizing a vector $\mathbf{v}$ by dividing it by its Euclidean norm, that is,

$$\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \tag{1.12}$$

where $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$ is the Euclidean norm of the vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$. In this way, the vector $\bar{\mathbf{v}}$ has length one and preserves the direction of $\mathbf{v}$.

1.3 Continuous convolution and discrete convolution

The convolution is a fundamental mathematical operation used to describe how an input signal is modified by a system. It is widely applied in the analysis of linear time-invariant (LTI) systems, signal processing, automatic control, and deep learning.

1.3.1 Continuous convolution

Given two real functions $f(t)$ and $g(t)$ defined on $\mathbb{R}$, their **continuous convolution** is defined as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t - \tau)\, d\tau. \tag{1.13}$$

In the context of LTI systems, if $f(t)$ represents the input to the system and $g(t)$ its impulse response, then $(f * g)(t)$ represents the output of the system.

1.3.2 Discrete convolution

When functions are represented as discrete-time sequences, convolution is defined as a weighted sum of products. Given two sequences $x[n]$ and $h[n]$, their discrete convolution is given by:

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n - k]. \tag{1.14}$$

In practice, if the sequences are finite and of bounded length, the sum is evaluated over a limited interval.

$$y[n] = \sum_{k=0}^{M-1} x[k]\, h[n-k], \quad \text{for } n = 0, 1, \ldots, M. \tag{1.15}$$

In both cases, continuous and discrete, convolution can be understood as an operation that measures the degree of overlap between a signal and a shifted version of another. This allows modeling the output of a system in response to any arbitrary input, as long as its impulse response is known.

1.4 Matrices

1.4.1 Identity matrix

The identity matrix $\mathbf{I}$ is a square matrix of size $n \times n$ whose diagonal elements are equal to 1 and the rest are 0. It is the neutral element in matrix multiplication, that is, for any matrix $\mathbf{A}$ of compatible size:

$$\mathbf{IA} = \mathbf{AI} = \mathbf{A}. \tag{1.16}$$

1.4.2 Matrix multiplication

Matrix multiplication is not commutative in general, but it is associative. For matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of compatible dimensions, we have:

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}) \quad \text{(Associativity)}, \tag{1.17}$$

but

$$\mathbf{AB} \neq \mathbf{BA} \quad \text{in general (Non-commutativity)}. \tag{1.18}$$

1.4.3 Augmented matrix

An augmented matrix is a matrix that combines two matrices, typically a coefficient matrix and a column of independent terms in a system of linear equations, for example:

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{array}\right]. \tag{1.19}$$

1.4.4 Inverse matrix

The inverse of a square matrix $\mathbf{A}$ is another matrix $\mathbf{A}^{-1}$ such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}. \tag{1.20}$$

It only exists if $\mathbf{A}$ is invertible (non-zero determinant).

Inverse of a matrix product

For invertible matrices $\mathbf{A}$ and $\mathbf{B}$, the reverse order property holds:

$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}. \tag{1.21}$$

Matrix transpose

The transpose of a matrix $\mathbf{A}$, denoted as $\mathbf{A}^\top$, is obtained by exchanging rows with columns:

$$(\mathbf{A}^\top)_{ij} = \mathbf{A}_{ji}. \tag{1.22}$$

Matrix rank

The *rank* of a matrix $\mathbf{A}$, denoted as $\text{rang}(\mathbf{A})$ or $\text{rank}(\mathbf{A})$, is the maximum number of linearly independent columns (or rows) of $\mathbf{A}$. It is equivalent to the number of non-zero pivots after applying Gaussian elimination (or row reduction).

Another way to understand it is as the dimension of the column space of $\mathbf{A}$, that is, the subspace spanned by its columns.

For example, if $\mathbf{A} \in \mathbb{R}^{m\times n}$, then:

$$\text{rank}(\mathbf{A}) \leq \min(m, n). \tag{1.23}$$

A matrix has **full rank** if its rank equals the minimum between its number of rows and columns. This is often an important condition for the existence and uniqueness of solutions in systems of linear equations.

Determinants and their properties

The *determinant* is a scalar value associated with a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, denoted as $\det(\mathbf{A})$ or $|\mathbf{A}|$. This number provides key information about the matrix, such as its invertibility, orientation, and volume scaling. The determinant of a matrix is zero when it has no inverse. The inverse

of a 2×2 square matrix, if it exists, can be computed using an explicit formula. Given the matrix

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \tag{1.24}$$

its inverse exists if and only if $\det(\mathbf{A}) \neq 0$, where

$$\det(\mathbf{A}) = ad - bc. \tag{1.25}$$

In that case, the inverse is given by:

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \operatorname{adj}(\mathbf{A}) = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{1.26}$$

where $\operatorname{adj}(\mathbf{A})$ is the transpose of the cofactor matrix of $\mathbf{A}$.

The eigenvalues of a matrix $\mathbf{A}$ are obtained by solving the characteristic equation:

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0, \tag{1.27}$$

where λ represents an eigenvalue and $\mathbf{I}$ is the identity matrix.

Determinant of a triangular matrix

If a square matrix $\mathbf{A}$ is **upper triangular** or **lower triangular**, its determinant is equal to the product of the elements on its main diagonal.

$$\det(\mathbf{A}) = a_{11} \cdot a_{22} \cdots a_{nn} = \prod_{i=1}^{n} a_{ii} \tag{1.28}$$

This property significantly simplifies the computation of the determinant, especially in elimination methods such as LU factorization or Gaussian elimination.

Example:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 4 & 5 \\ 0 & 0 & -1 \end{bmatrix} \quad \Rightarrow \quad \det(\mathbf{A}) = 2 \cdot 4 \cdot (-1) = -8 \tag{1.29}$$

1.5 Eigenvalues and Eigenvectors

Given a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, an **eigenvector** (or characteristic vector) $\mathbf{x} \neq \mathbf{0}$ and an **eigenvalue** $\lambda \in \mathbb{R}$ satisfy the equation:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{1.30}$$

This means that multiplying the matrix $\mathbf{A}$ by the vector $\mathbf{x}$ results in the same vector scaled by the scalar λ.

This property has an important computational use: if $\mathbf{x}$ is an eigenvector of $\mathbf{A}$, then:

$$\mathbf{A}^k\mathbf{x} = \lambda^k\mathbf{x}, \quad \text{for all } k \in \mathbb{N}. \tag{1.31}$$

That is, raising $\mathbf{A}$ to a power and applying it to $\mathbf{x}$ is simply equivalent to scaling $\mathbf{x}$ by λ^k, which is much more computationally efficient.

Eigenvalues are obtained by solving the characteristic equation:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0. \tag{1.32}$$

The number of eigenvalues of a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ (counting multiplicities) equals the number of roots of the characteristic polynomial

$$p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0, \tag{1.33}$$

That is, there will be n eigenvalues, some of which may repeat according to their algebraic multiplicity.

If $\mathbf{x}$ are the eigenvectors corresponding to the eigenvalue λ, then:

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}, \tag{1.34}$$

Therefore, by solving the previous equation for each eigenvalue, we find the eigenvector $\mathbf{x}$ associated with that eigenvalue.

Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ be a square matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ (counting multiplicities). Then:

$$\prod_{i=1}^{n} \lambda_i = \det(\mathbf{A}) \tag{1.35}$$

and

$$\sum_{i=1}^{n} \lambda_i = \operatorname{tr}(\mathbf{A}), \tag{1.36}$$

where $\det(\mathbf{A})$ is the determinant of $\mathbf{A}$ and $\mathrm{tr}(\mathbf{A})$ is the trace of $\mathbf{A}$, that is, the sum of the elements on its main diagonal.

Note 1: In **MATLAB®**, we can compute eigenvalues and eigenvectors using the command `[V, D] = eig(A)`, where `D` is a diagonal matrix containing the eigenvalues, and `V` contains the eigenvectors as columns; that is, $A = VDV^{-1}$ if A is diagonalizable.

1.6 Jordan Normal Form

Given a square matrix $A \in \mathbb{C}^{n\times n}$, there exists an invertible matrix P such that:

$$J = PAP^{-1} = \begin{bmatrix} J_1 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & J_3 & 0 \\ 0 & 0 & 0 & J_{\mathbf{P}} \end{bmatrix} \tag{1.37}$$

where J is the **Jordan canonical form** of A, and J_k is a special block called a *Jordan block*.

1.6.1 Jordan block

A Jordan block $J_k(\lambda)$ associated with the eigenvalue λ and of size $n \times n$ has the form:

$$J_k(\lambda) = \begin{bmatrix} \lambda_k & 1 & 0 & \cdots & 0 \\ 0 & \lambda_k & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_k & 1 \\ 0 & \cdots & \cdots & 0 & \lambda_k \end{bmatrix}_{n_k \times n_k} \tag{1.38}$$

Here, each λ_k is an eigenvalue of A, and the number $\mathbf{P}$ of Jordan blocks equals the total number of linearly independent eigenvectors of A. The matrix J is unique and is called the Jordan normal form of A.

Note 2: Using **MATLAB®**, it is possible to compute the Jordan canonical form of a square matrix A with the command `[J] = jordan(A)`.

1.7 Singular Value Decomposition (SVD)

When the matrix $\mathbf{A}$ is not square, the direct computation of eigenvalues is not possible, since eigenvalues are only defined for square matrices. However, it is still possible to compute the *singular values* of a rectangular matrix.

The Singular Value Decomposition, known as **SVD**, of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ allows us to factorize it as the product of three matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{1.39}$$

where:

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix whose columns are the *left singular vectors* of $\mathbf{A}$.
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix whose columns are the *right singular vectors*.
- $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a (possibly rectangular) diagonal matrix with the *singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ on its diagonal.

$$\mathbf{A} = \mathbf{u}_1 \sigma_1 \mathbf{v}_1^\top + \mathbf{u}_2 \sigma_2 \mathbf{v}_2^\top + \cdots + \mathbf{u}_r \sigma_r \mathbf{v}_r^\top \tag{1.40}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i-th columns of the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, respectively, and σ_i is the i-th singular value, corresponding to a diagonal entry of $\mathbf{\Sigma}$.

Each singular value σ_i is the square root of an eigenvalue of the symmetric matrix $\mathbf{A}^\top \mathbf{A}$, that is:

$$\sigma_i = \sqrt{\lambda_i}, \quad \text{where } \lambda_i \text{ is an eigenvalue of } \mathbf{A}^\top \mathbf{A}. \tag{1.41}$$

The SVD is particularly useful for:

- Matrix approximation (rank reduction).
- Solving overdetermined or ill-conditioned systems.
- Understanding internal data structures in multivariate analysis (e.g., PCA).

Note 3: Using **MATLAB®**, singular values of a matrix A can be computed with the command `S = svd(A)`, where S contains the singular values sorted in descending order. The full decomposition can also be obtained with `[U, S, V] = svd(A)`.

Moore-Penrose Pseudoinverse

When a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is not square or not invertible, it is not possible to compute a classical inverse. In such cases, the **Moore-Penrose pseudoinverse**, denoted by $\mathbf{A}^+$, generalizes the concept of inverse for rectangular matrices.

The pseudoinverse can be computed from the Singular Value Decomposition (SVD) of the matrix:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{1.42}$$

then the pseudoinverse is given by:

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^\top, \tag{1.43}$$

where $\mathbf{\Sigma}^+$ is obtained by inverting the nonzero singular values in $\mathbf{\Sigma}$, transposing the matrix, and filling the rest with zeros.

The pseudoinverse has many applications, including dimensionality reduction and regularization in machine learning and data processing.

Note 4: Using **MATLAB®**, the *Moore-Penrose pseudoinverse* of a matrix A, square or rectangular, can be computed with the command `A_pinv = pinv(A)`.

Matrix Norm

The norm of a matrix $\mathbf{A}$, understood as its **induced 2-norm** or **spectral norm**, is defined as:

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}. \tag{1.44}$$

If the matrix $\mathbf{A}$ is not symmetric, its induced 2-norm can be computed from the symmetric positive semidefinite matrix $\mathbf{A}^\top\mathbf{A}$, since it preserves the necessary information about the singular values of $\mathbf{A}$. In particular:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}, \tag{1.45}$$

where $\lambda_{\max}(\mathbf{A}^\top \mathbf{A})$ represents the largest eigenvalue of the symmetric matrix $\mathbf{A}^\top \mathbf{A}$. This square root equals the maximum singular value of $\mathbf{A}$, ensuring that:

$$\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A}). \tag{1.46}$$

Note 5: Using **MATLAB®**, different *matrix norms* can be computed with the command `norm(A, p)`, where A is a matrix and p specifies the type of norm. For example: `norm(A, 2)` computes the *spectral norm* (largest singular value), `norm(A, 'fro')` the *Frobenius norm*, and `norm(A, 1)` or `norm(A, inf)` the induced 1-norm and infinity-norm, respectively.

1.8 Complex Numbers

The transfer functions of a dynamic system, whether in discrete or continuous time, as well as the eigenvalues obtained from the state-transition matrix of a system, are often represented using complex numbers of the form $\gamma = a + bi$. In this expression, i denotes the imaginary unit, which satisfies the fundamental property $i^2 = -1$. It is worth noting that in electrical engineering it is common to use j instead of i to avoid confusion with the notation for electric current Strang [2016]. This imaginary number is deeply connected to important identities such as Euler's formula, where $e^{2\pi i} = 1$, revealing the inherent periodicity in the complex plane. The modulus of the complex number is given by $|\gamma| = \sqrt{a^2 + b^2}$.

The concept of complex numbers is essential in system analysis. On one hand, in the Laplace Transform, the variable s is defined as a complex number $s = \sigma + \omega i$, where σ represents the real part (associated with system damping) and ω corresponds to the angular frequency in radians per second. On the other hand, in discrete systems we work with the Z Transform, where the variable z is directly related to s through the sampling period T via the expression $z = e^{sT} = e^{(\sigma+\omega i)T} = e^{\sigma T} \cdot e^{\omega T i}$. In this formulation, the term $e^{\sigma T}$ determines the magnitude behavior, while $e^{\omega T i}$ provides information about the phase, elegantly connecting the continuous and discrete worlds.

A complex number $\gamma = a + bi$ has a corresponding complex conjugate defined as $\bar{\gamma} = a - bi$. Its inverse is given by $\frac{1}{\gamma} = \frac{\bar{\gamma}}{|\gamma|^2} = \frac{\bar{\gamma}}{a^2+b^2}$.

Consider a column vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)^\top$ in the space $\mathbb{C}^n$ of n-tuples of complex numbers. The squared norm of $\boldsymbol{\gamma}$ is defined as the sum of the squares of the moduli of its components:

$$\|\boldsymbol{\gamma}\|^2 = \sum_{k=1}^{n} |\gamma_k|^2 = |\gamma_1|^2 + |\gamma_2|^2 + \cdots + |\gamma_n|^2, \tag{1.47}$$

where each $|\gamma_k|$ represents the modulus of the complex number γ_k, given by $|\gamma_k| = \sqrt{\gamma_k \overline{\gamma_k}}$. This definition coincides with the canonical inner product in $\mathbb{C}^n$ and generalizes the notion of Euclidean length to the complex case. Note that when all γ_k are real, we recover the classical formula for vectors in $\mathbb{R}^n$.

For a complex vector $\boldsymbol{\gamma} \in \mathbb{C}^n$, its squared norm is obtained via the product $\overline{\boldsymbol{\gamma}}^\top \boldsymbol{\gamma}$, where $\overline{\boldsymbol{\gamma}}^\top$ represents the transpose of the conjugate vector. This product, which equals the sum $\sum_{k=1}^{n} \overline{\gamma_k}\gamma_k = \sum_{k=1}^{n} |\gamma_k|^2$, always yields a non-negative real scalar.

1.9 Moments and Central Moments

Moments are statistical measures that describe fundamental characteristics of a random variable. Given a random variable X and a positive integer r, the r-th order moment is defined as the expected value of the r-th power of the variable:

$$\mu'_r = \mathbb{E}[X^r], \tag{1.48}$$

where μ'_r is known as the r-th order moment about the origin. It describes properties such as central tendency ($r = 1$), dispersion ($r = 2$), skewness ($r = 3$), and kurtosis ($r = 4$), among others.

1.9.1 Central Moments

Central moments are a generalization that measures the powers of deviations from the mean. The r-th order central moment is defined as:

$$\mu_r = \mathbb{E}[(X - \mathbb{E}[X])^r], \tag{1.49}$$

where μ_r represents the r-th order central moment. In particular:

- $\mu_1 = 0$, since the mean of deviations from the mean is zero.
- $\mu_2 = \mathbb{V}[X]$ is the variance, measuring data dispersion.
- μ_3 indicates skewness: if positive, the distribution has a right tail; if negative, a left tail.

- μ_4 relates to kurtosis, describing the concentration of values around the mean and in the tails.

1.9.2 Relationship Between Moments and Central Moments

There is a relationship between central moments and moments about the origin. For example, the second central moment (variance) can be written as:

$$\mu_2 = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \mu_2' - (\mu_1')^2. \tag{1.50}$$

This relationship allows expressing central moments in terms of raw moments and is useful for theoretical and computational analysis.

1.9.3 Interpretation as Outer Products

Central moments can be interpreted as outer products in the space of random variables, particularly in terms of covariances. This interpretation reveals an algebraic structure useful for statistical analysis and vectorial treatment of random variables.

The second central moment, or variance, can be expressed as:

$$\mu_2 = \mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2], \tag{1.51}$$

and represents the outer product of X with itself, in the sense of measuring dispersion around its mean.

More generally, the **outer product between two random variables** X and Y is given by their **covariance**:

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])], \tag{1.52}$$

which can be understood as a measure of the linear interaction between deviations of both variables from their means. This expression is symmetric, bilinear, and plays a role analogous to an outer product in multilinear algebra.

In matrix form, if we consider n random variables as a random vector $\mathbf{X} = [X_1, X_2, \ldots, X_n]^T$, the second-order tensor given by:

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^T\right] \tag{1.53}$$

corresponds to the covariance matrix, which generalizes the idea of an outer product to higher-dimensional spaces, and is central in multivariate statistical analysis.

This approach provides a solid foundation for constructing orthogonal decompositions, correlation analysis, dimensionality reduction (such as PCA), and geometric formulations of statistics.

Note 6: In **MATLAB®**, central moments can be calculated manually or using functions such as `moment(x, n)` from the Statistics and Machine Learning Toolbox.

1.10 From Laplace to Fourier: a Connection on the Imaginary Axis

The Laplace transform is one of the fundamental tools in control system analysis and design, as it allows describing the dynamics of a system in the complex domain using the variable $s = \sigma + j\omega$.

Laplace Transform

The Laplace transform of a signal $x(t)$ is defined as:

$$X(s) = \mathcal{L}\{x(t)\} \triangleq \int_{0^-}^{\infty} x(t)\, e^{-st}\, dt, \qquad s = \sigma + j\omega. \tag{1.54}$$

Inverse Laplace Transform

The inverse Laplace transform allows recovering the original signal in the time domain:

$$x(t) = \mathcal{L}^{-1}\{X(s)\} \triangleq \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} (X(s)\, e^{st})|_{s=\sigma+j\omega}\, d\omega, \tag{1.55}$$

Example: Inverse Transform of a Third-Order System

Consider a third-order linear system with a general transfer function:

$$H(s) = \frac{b_0 s^2 + b_1 s + b_2}{(s - p_1)(s - p_2)(s - p_3)}, \tag{1.56}$$

where p_1, p_2, p_3 are the system poles, which may be real, complex, or complex conjugates, and b_0, b_1, b_2 are the numerator coefficients, determining the system zeros.

Using partial fraction decomposition, it can be expressed as:

$$H(s) = \frac{A}{s - p_1} + \frac{B}{s - p_2} + \frac{C}{s - p_3}, \tag{1.57}$$

where the coefficients A, B, C depend on the poles and numerator coefficients.

Applying the inverse Laplace transform, the system's time-domain response is:

$$h(t) = \mathcal{L}^{-1}\{H(s)\} = A\, e^{p_1 t} + B\, e^{p_2 t} + C\, e^{p_3 t}, \quad t \geq 0. \tag{1.58}$$

Each pole p_i in the Laplace domain contributes an exponential term $e^{p_i t}$ in the time domain, and the coefficients A, B, C modulated by the zeros determine the amplitude of each component. Thus, the full time-domain dynamics of the system are defined both by the pole locations and the numerator coefficients of the transfer function.

We know that the poles belong to the complex domain $p \in \mathbb{C} \rightarrow p = \sigma + j\omega$, and the associated exponential term in the time response is therefore:

$$e^{pt} = e^{(\sigma + j\omega)t} = e^{\sigma t}\, e^{j\omega t}. \tag{1.59}$$

Applying Euler's formula, we can separate the real and imaginary parts:

$$\mathrm{Re}\{e^{pt}\} = e^{\sigma t}\cos(\omega t), \qquad \mathrm{Im}\{e^{pt}\} = e^{\sigma t}\sin(\omega t). \tag{1.60}$$

Thus, a complex pole generates a damped oscillatory term in the time response, where its real part σ determines the decay or growth rate and ω the oscillation frequency.

Since the evolution of the time response $h(t)$ is determined by the exponential term e^{pt}, both the real and imaginary parts of the response grow or decay depending on the real part of the pole σ. From this observation, one of the most important conclusions about the stability of a continuous-time dynamic system defined by its transfer function follows:

- If the real part of all poles is negative ($\sigma < 0$), the time response decays to zero (stable).

- If the real part of any pole is positive ($\sigma > 0$), the time response grows without bound (unstable).

- If the real part is exactly zero ($\sigma = 0$), the response remains constant in magnitude, producing sustained oscillations in the case of complex poles (marginally stable).

Consequently, the input-output stability of a continuous linear system is completely determined by the location of the real component of the poles in the complex plane.

Fourier Transform

The Fourier transform of a signal $x(t)$ is defined as:

$$X(j\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt. \tag{1.61}$$

Its inverse transform allows recovering the original signal in the time domain:

$$x(t) = \mathcal{F}^{-1}\{X(j\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)\, e^{j\omega t}\, d\omega. \tag{1.62}$$

Relationship Between Both

Notice that the Fourier transform 1.61 is similar to the Laplace transform 1.54, obtained by evaluating the complex variable s on the imaginary axis jw, with a clear difference in the integration limits of the two equations.

When the signal $x(t)$ is real, its Fourier transform satisfies the *conjugate symmetry* property:

$$X(-j\omega) = X(j\omega), \tag{1.63}$$

which reflects symmetry in magnitude and phase of the spectral components.

In this way, the Laplace transform provides a more general framework, while the Fourier transform focuses on spectral analysis, particularly useful when the signal has finite energy or is stationary.

1.11 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) is a fundamental tool for the spectral analysis of discrete-time signals. It allows representing a finite sequence of N samples in the frequency domain as a combination of complex sinusoids.

Given a discrete-time signal $x[n]$, defined for $n = 0, 1, \ldots, N-1$, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1, \tag{1.64}$$

where:

- $X[k]$ represents the spectral value of the sequence at discrete frequency k.
- The factor $e^{-j\frac{2\pi}{N}kn}$ corresponds to the orthogonal basis functions in Fourier space.

The inverse transform, known as the IDFT (*Inverse Discrete Fourier Transform*), allows recovering the signal in the time domain:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \ldots, N-1. \tag{1.65}$$

In this way, the DFT establishes a bidirectional relationship between the time domain and the frequency domain for finite discrete signals, forming the theoretical foundation of computational algorithms such as the *Fast Fourier Transform* (FFT).

1.12 Definition of a Linear System

A dynamic system is called **linear** if it satisfies the *superposition principle*, which encompasses two fundamental properties:

- **Additivity:** if an input $u_1(t)$ produces output $y_1(t)$, and another input $u_2(t)$ produces output $y_2(t)$, then the combined input $u_1(t) + u_2(t)$ produces the output $y_1(t) + y_2(t)$.
- **Homogeneity:** if an input $u(t)$ produces output $y(t)$, then a scaled input $\alpha u(t)$, with $\alpha \in \mathbb{R}$, produces the scaled output $\alpha y(t)$.

In a compact form, the superposition principle states that, for any linear combination of inputs, the corresponding output is the same linear combination of the individual outputs:

$$\alpha_1 u_1(t) + \alpha_2 u_2(t) \quad \longrightarrow \quad \alpha_1 y_1(t) + \alpha_2 y_2(t), \quad \forall\, \alpha_1, \alpha_2 \in \mathbb{R}. \tag{1.66}$$

Therefore, a system is linear if and only if it simultaneously satisfies additivity and homogeneity, properties that together constitute the superposition principle.

Most engineering processes can be approximated or represented by an equivalent linear system. This representation is highly relevant, as linear systems, in contrast to nonlinear systems, possess a well-developed theoretical framework that allows rigorous and relatively straightforward analysis. Consequently, the study of linear models is a fundamental tool in engineering, not only because of their practical utility in system design and analysis, but also because they provide a solid conceptual foundation upon which extensions to nonlinear system analysis can be built. In practice, real systems are *bounded input, bounded output* (BIBO), meaning that the inputs and outputs are inherently limited, which ensures predictable and stable behavior within operating ranges.

1.13 LaSalle's Invariance Principle

LaSalle's Invariance Principle (also known as LaSalle's Theorem) is a generalization and extension of the direct Lyapunov method for analyzing the stability of autonomous dynamical systems. This principle is particularly powerful for proving asymptotic stability in cases where the derivative of the candidate Lyapunov function is only negative semidefinite (i.e., $\dot{V}(\mathbf{x}) \leq 0$), which by itself is insufficient to conclude global asymptotic stability using the classical Lyapunov theorem.

Consider the autonomous system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, with $\mathbf{f}$ a locally Lipschitz function, and let $\Omega \subset D$ be a compact set (closed and bounded) that is positively invariant with respect to the system. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \Omega$.

Define E as the set of points in Ω where $\dot{V}(\mathbf{x}) = 0$:

$$E = \left\{ \mathbf{x} \in \Omega \mid \dot{V}(\mathbf{x}) = 0 \right\} \tag{1.67}$$

and let M be the largest invariant set contained in E. LaSalle's Invariance Principle then states that every trajectory starting in Ω converges to M

as $t \to \infty$:

$$\mathbf{x}(t) \to M \quad \text{as} \quad t \to \infty \tag{1.68}$$

In practice, this theorem allows one to demonstrate convergence to equilibrium by identifying the largest invariant set within the region where the system's energy (or Lyapunov function) stops decreasing.

1.14 Barbalat's Lemma

Barbalat's Lemma is a fundamental result in the analysis of non-autonomous (time-varying) systems that relates the properties of a function and its derivative to conclude convergence to zero. Unlike Lyapunov theorems for autonomous systems, Barbalat's Lemma is essential for addressing asymptotic stability in systems where the candidate Lyapunov function $V(t, \mathbf{x})$ depends explicitly on time.

A widely used corollary in control theory is the following:

- **Corollary:** If a function $f(t)$ is differentiable and has a finite limit as $t \to \infty$, and if its derivative $\dot{f}(t)$ is uniformly continuous, then $\dot{f}(t) \to 0$ as $t \to \infty$.

The uniform continuity condition of $f(t)$ is often verified by showing that its derivative, $\dot{f}(t)$, is bounded. This lemma is a key tool to demonstrate that a signal (such as a tracking error $e(t)$) converges asymptotically to zero in stability analysis of non-autonomous systems, even when a Lyapunov function with a negative definite derivative cannot be found.

1.15 Lyapunov Stability Theorem

Consider a dynamical system such that:

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n, \quad f(0) = 0, \tag{1.69}$$

and let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that:

1. $V(0) = 0$ and $V(x) > 0$ for all $x \neq 0$ (positive definite),
2. $\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \leq 0$ for all x (negative semidefinite).

Then, the equilibrium point $x = 0$ is stable.

Moreover, if $\dot{V}(x) < 0$ for all $x \neq 0$ (negative definite), then $x = 0$ is asymptotically stable.