Appendices

A

Linear Algebra

Since modelling and control of robot manipulators requires an extensive use of *matrices* and *vectors* as well as of matrix and vector *operations*, the goal of this appendix is to provide a brush-up of *linear algebra*.

A.1 Definitions

A *matrix* of dimensions $(m \times n)$, with m and n positive integers, is an array of elements a_{ij} arranged into m *rows* and n *columns*:

$$\boldsymbol{A} = [a_{ij}]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \tag{A.1}$$

If $m = n$, the matrix is said to be *square*; if $m < n$, the matrix has more columns than rows; if $m > n$ the matrix has more rows than columns. Further, if $n = 1$, the notation (A.1) is used to represent a (column) vector $\boldsymbol{a}$ of dimensions $(m \times 1)$;[1] the elements a_i are said to be vector components.

A square matrix $\boldsymbol{A}$ of dimensions $(n \times n)$ is said to be *upper triangular* if $a_{ij} = 0$ for $i > j$:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix};$$

the matrix is said to be *lower triangular* if $a_{ij} = 0$ for $i < j$.

---

[1] According to standard mathematical notation, small boldface is used to denote vectors while capital boldface is used to denote matrices. Scalars are denoted by roman characters.

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *diagonal* if $a_{ij} = 0$ for $i \neq j$, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} = \text{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}.$$

If an $(n \times n)$ diagonal matrix has all unit elements on the diagonal ($a_{ii} = 1$), the matrix is said to be *identity* and is denoted by $\boldsymbol{I}_n$.[2] A matrix is said to be *null* if all its elements are null and is denoted by $\boldsymbol{O}$. The null column vector is denoted by $\mathbf{0}$.

The *transpose* $\boldsymbol{A}^T$ of a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ is the matrix of dimensions $(n \times m)$ which is obtained from the original matrix by interchanging its rows and columns:

$$\boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}. \tag{A.2}$$

The transpose of a column vector $\boldsymbol{a}$ is the row vector $\boldsymbol{a}^T$.

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *symmetric* if $\boldsymbol{A}^T = \boldsymbol{A}$, and thus $a_{ij} = a_{ji}$:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{12} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{nn} \end{bmatrix}.$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *skew-symmetric* if $\boldsymbol{A}^T = -\boldsymbol{A}$, and thus $a_{ij} = -a_{ji}$ for $i \neq j$ and $a_{ii} = 0$, leading to

$$\boldsymbol{A} = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ -a_{12} & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{1n} & -a_{2n} & \dots & 0 \end{bmatrix}.$$

A *partitioned* matrix is a matrix whose elements are matrices (*blocks*) of proper dimensions:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \dots & \boldsymbol{A}_{1n} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \dots & \boldsymbol{A}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{m1} & \boldsymbol{A}_{m2} & \dots & \boldsymbol{A}_{mn} \end{bmatrix}.$$

[2] Subscript n is usually omitted if the dimensions are clear from the context.

A partitioned matrix may be block-triangular or block-diagonal. Special partitions of a matrix are that by columns

$$\boldsymbol{A} = [\,\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \dots \quad \boldsymbol{a}_n\,]$$

and that by rows

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix}.$$

Given a square matrix $\boldsymbol{A}$ of dimensions $(n \times n)$, the *algebraic complement* $\boldsymbol{A}_{(ij)}$ of element a_{ij} is the matrix of dimensions $((n-1) \times (n-1))$ which is obtained by eliminating row i and column j of matrix $\boldsymbol{A}$.

A.2 Matrix Operations

The *trace* of an $(n \times n)$ square matrix $\boldsymbol{A}$ is the sum of the elements on the diagonal:

$$\mathrm{Tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}. \tag{A.3}$$

Two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same dimensions $(m \times n)$ are equal if $a_{ij} = b_{ij}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are two matrices of the same dimensions, their *sum* is the matrix

$$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B} \tag{A.4}$$

whose elements are given by $c_{ij} = a_{ij} + b_{ij}$. The following properties hold:

$$\begin{aligned} &\boldsymbol{A} + \boldsymbol{O} = \boldsymbol{A} \\ &\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A} \\ &(\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C}). \end{aligned}$$

Notice that two matrices of the same dimensions and partitioned in the same way can be summed formally by operating on the blocks in the same position and treating them like elements.

The *product of a scalar* α *by an* $(m \times n)$ *matrix* $\boldsymbol{A}$ is the matrix $\alpha\boldsymbol{A}$ whose elements are given by αa_{ij}. If $\boldsymbol{A}$ is an $(n \times n)$ diagonal matrix with all equal elements on the diagonal $(a_{ii} = a)$, it follows that $\boldsymbol{A} = a\boldsymbol{I}_n$.

If $\boldsymbol{A}$ is a square matrix, one may write

$$\boldsymbol{A} = \boldsymbol{A}_s + \boldsymbol{A}_a \tag{A.5}$$

where

$$\boldsymbol{A}_s = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T) \tag{A.6}$$

is a symmetric matrix representing the *symmetric* part of $\boldsymbol{A}$, and

$$\boldsymbol{A}_a = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^T) \tag{A.7}$$

is a skew-symmetric matrix representing the *skew-symmetric* part of $\boldsymbol{A}$.

The row-by-column *product* of a matrix $\boldsymbol{A}$ of dimensions $(m \times p)$ by a matrix $\boldsymbol{B}$ of dimensions $(p \times n)$ is the matrix of dimensions $(m \times n)$

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B} \tag{A.8}$$

whose elements are given by $c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$. The following properties hold:

$$\begin{aligned}
&\boldsymbol{A} = \boldsymbol{A}\boldsymbol{I}_p = \boldsymbol{I}_m \boldsymbol{A}\\
&\boldsymbol{A}(\boldsymbol{B}\boldsymbol{C}) = (\boldsymbol{A}\boldsymbol{B})\boldsymbol{C}\\
&\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{A}\boldsymbol{B} + \boldsymbol{A}\boldsymbol{C}\\
&(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{C} = \boldsymbol{A}\boldsymbol{C} + \boldsymbol{B}\boldsymbol{C}\\
&(\boldsymbol{A}\boldsymbol{B})^T = \boldsymbol{B}^T \boldsymbol{A}^T.
\end{aligned}$$

Notice that, in general, $\boldsymbol{A}\boldsymbol{B} \neq \boldsymbol{B}\boldsymbol{A}$, and $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{O}$ does not imply that $\boldsymbol{A} = \boldsymbol{O}$ or $\boldsymbol{B} = \boldsymbol{O}$; further, notice that $\boldsymbol{A}\boldsymbol{C} = \boldsymbol{B}\boldsymbol{C}$ does not imply that $\boldsymbol{A} = \boldsymbol{B}$.

If an $(m \times p)$ matrix $\boldsymbol{A}$ and a $(p \times n)$ matrix $\boldsymbol{B}$ are partitioned in such a way that the number of blocks for each row of $\boldsymbol{A}$ is equal to the number of blocks for each column of $\boldsymbol{B}$, and the blocks $\boldsymbol{A}_{ik}$ and $\boldsymbol{B}_{kj}$ have dimensions compatible with product, the matrix product $\boldsymbol{A}\boldsymbol{B}$ can be formally obtained by operating by rows and columns on the blocks of proper position and treating them like elements.

For an $(n \times n)$ *square* matrix $\boldsymbol{A}$, the *determinant* of $\boldsymbol{A}$ is the scalar given by the following expression, which holds $\forall i = 1, \ldots, n$:

$$\det(\boldsymbol{A}) = \sum_{j=1}^{n} a_{ij} (-1)^{i+j} \det\big(\boldsymbol{A}_{(ij)}\big). \tag{A.9}$$

The determinant can be computed according to any row i as in (A.9); the same result is obtained by computing it according to any column j. If $n = 1$, then $\det(a_{11}) = a_{11}$. The following property holds:

$$\det(\boldsymbol{A}) = \det(\boldsymbol{A}^T).$$

Moreover, interchanging two generic columns p and q of a matrix $\boldsymbol{A}$ yields

$$\det\big([\,\boldsymbol{a}_1 \ldots \boldsymbol{a}_p \ldots \boldsymbol{a}_q \ldots \boldsymbol{a}_n\,]\big) = -\det\big([\,\boldsymbol{a}_1 \ldots \boldsymbol{a}_q \ldots \boldsymbol{a}_p \ldots \boldsymbol{a}_n\,]\big).$$

As a consequence, if a matrix has two equal columns (rows), then its determinant is null. Also, it is $\det(\alpha \boldsymbol{A}) = \alpha^n \det(\boldsymbol{A})$.

Given an $(m \times n)$ matrix $\boldsymbol{A}$, the determinant of the square block obtained by selecting an equal number k of rows and columns is said to be k-order *minor* of matrix $\boldsymbol{A}$. The minors obtained by taking the *first* k rows and columns of $\boldsymbol{A}$ are said to be *principal* minors.

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices, then

$$\det(\boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{A})\det(\boldsymbol{B}). \tag{A.10}$$

If $\boldsymbol{A}$ is an $(n \times n)$ triangular matrix (in particular diagonal), then

$$\det(\boldsymbol{A}) = \prod_{i=1}^{n} a_{ii}.$$

More generally, if $\boldsymbol{A}$ is block-triangular with m blocks $\boldsymbol{A}_{ii}$ on the diagonal, then

$$\det(\boldsymbol{A}) = \prod_{i=1}^{m} \det(\boldsymbol{A}_{ii}).$$

A square matrix $\boldsymbol{A}$ is said to be *singular* when $\det(\boldsymbol{A}) = 0$.

The *rank* $\varrho(\boldsymbol{A})$ of a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ is the maximum integer r so that at least a non-null minor of order r exists. The following properties hold:

$$\begin{aligned}
&\varrho(\boldsymbol{A}) \le \min\{m, n\}\\
&\varrho(\boldsymbol{A}) = \varrho(\boldsymbol{A}^T)\\
&\varrho(\boldsymbol{A}^T\boldsymbol{A}) = \varrho(\boldsymbol{A})\\
&\varrho(\boldsymbol{A}\boldsymbol{B}) \le \min\{\varrho(\boldsymbol{A}), \varrho(\boldsymbol{B})\}.
\end{aligned}$$

A matrix so that $\varrho(\boldsymbol{A}) = \min\{m, n\}$ is said to be *full-rank*.

The *adjoint* of a square matrix $\boldsymbol{A}$ is the matrix

$$\operatorname{Adj}\boldsymbol{A} = [(-1)^{i+j}\det(\boldsymbol{A}_{(ij)})]^T_{\substack{i=1,\dots,n\\ j=1,\dots,n}}. \tag{A.11}$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *invertible* if a matrix $\boldsymbol{A}^{-1}$ exists, termed *inverse* of $\boldsymbol{A}$, so that

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}_n.$$

Since $\varrho(\boldsymbol{I}_n) = n$, an $(n \times n)$ square matrix $\boldsymbol{A}$ is invertible if and only if $\varrho(\boldsymbol{A}) = n$, i.e., $\det(\boldsymbol{A}) \ne 0$ (nonsingular matrix). The inverse of $\boldsymbol{A}$ can be computed as

$$\boldsymbol{A}^{-1} = \frac{1}{\det(\boldsymbol{A})}\operatorname{Adj}\boldsymbol{A}. \tag{A.12}$$

The following properties hold:

$$\begin{aligned}
&(\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A}\\
&(\boldsymbol{A}^T)^{-1} = (\boldsymbol{A}^{-1})^T.
\end{aligned}$$

If the inverse of a square matrix is equal to its transpose

$$\boldsymbol{A}^T = \boldsymbol{A}^{-1} \tag{A.13}$$

then the matrix is said to be *orthogonal*; in this case it is

$$\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}. \tag{A.14}$$

A square matrix $\boldsymbol{A}$ is said *idempotent* if

$$\boldsymbol{A}\boldsymbol{A} = \boldsymbol{A}. \tag{A.15}$$

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are invertible square matrices of the same dimensions, then

$$(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}. \tag{A.16}$$

Given n square matrices $\boldsymbol{A}_{ii}$ all invertible, the following expression holds:

$$\big(\mathrm{diag}\{\boldsymbol{A}_{11}, \dots, \boldsymbol{A}_{nn}\}\big)^{-1} = \mathrm{diag}\{\boldsymbol{A}_{11}^{-1}, \dots, \boldsymbol{A}_{nn}^{-1}\}.$$

where $\mathrm{diag}\{\boldsymbol{A}_{11}, \dots, \boldsymbol{A}_{nn}\}$ denotes the block-diagonal matrix.

If $\boldsymbol{A}$ and $\boldsymbol{C}$ are invertible square matrices of proper dimensions, the following expression holds:

$$(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{C}\boldsymbol{D})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{C}^{-1})^{-1}\boldsymbol{D}\boldsymbol{A}^{-1},$$

where the matrix $\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{C}^{-1}$ must be invertible.

If a block-partitioned matrix is invertible, then its inverse is given by the general expression

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{D} \\ \boldsymbol{C} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \boldsymbol{E}\boldsymbol{\Delta}^{-1}\boldsymbol{F} & -\boldsymbol{E}\boldsymbol{\Delta}^{-1} \\ -\boldsymbol{\Delta}^{-1}\boldsymbol{F} & \boldsymbol{\Delta}^{-1} \end{bmatrix} \tag{A.17}$$

where $\boldsymbol{\Delta} = \boldsymbol{B} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{D}$, $\boldsymbol{E} = \boldsymbol{A}^{-1}\boldsymbol{D}$ and $\boldsymbol{F} = \boldsymbol{C}\boldsymbol{A}^{-1}$, under the assumption that the inverses of matrices $\boldsymbol{A}$ and $\boldsymbol{\Delta}$ exist. In the case of a block-triangular matrix, invertibility of the matrix requires invertibility of the blocks on the diagonal. The following expressions hold:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{C} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} & \boldsymbol{O} \\ -\boldsymbol{B}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{B}^{-1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{D} \\ \boldsymbol{O} & \boldsymbol{B} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{D}\boldsymbol{B}^{-1} \\ \boldsymbol{O} & \boldsymbol{B}^{-1} \end{bmatrix}.$$

The *derivative* of an $(m \times n)$ matrix $\boldsymbol{A}(t)$, whose elements $a_{ij}(t)$ are differentiable functions, is the matrix

$$\dot{\boldsymbol{A}}(t) = \frac{d}{dt}\boldsymbol{A}(t) = \left[\frac{d}{dt}a_{ij}(t)\right]_{\substack{i=1,\dots,m \\ j=1,\dots,n}}. \tag{A.18}$$

If an $(n \times n)$ square matrix $\boldsymbol{A}(t)$ is so that $\varrho(\boldsymbol{A}(t)) = n \; \forall t$ and its elements $a_{ij}(t)$ are differentiable functions, then the derivative of the *inverse* of $\boldsymbol{A}(t)$ is given by

$$\frac{d}{dt}\boldsymbol{A}^{-1}(t) = -\boldsymbol{A}^{-1}(t)\dot{\boldsymbol{A}}(t)\boldsymbol{A}^{-1}(t). \tag{A.19}$$

Given a scalar function $f(\boldsymbol{x})$, endowed with partial derivatives with respect to the elements x_i of the $(n \times 1)$ vector $\boldsymbol{x}$, the *gradient* of function f with respect to vector $\boldsymbol{x}$ is the $(n \times 1)$ column vector

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T = \begin{bmatrix} \dfrac{\partial f(\boldsymbol{x})}{\partial x_1} & \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} & \dots & \dfrac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix}^T. \tag{A.20}$$

Further, if $\boldsymbol{x}(t)$ is a differentiable function with respect to t, then

$$\dot{f}(\boldsymbol{x}) = \frac{d}{dt} f(\boldsymbol{x}(t)) = \frac{\partial f}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \nabla_{\boldsymbol{x}}^T f(\boldsymbol{x})\dot{\boldsymbol{x}}. \tag{A.21}$$

Given a vector function $\boldsymbol{g}(\boldsymbol{x})$ of dimensions $(m \times 1)$, whose elements g_i are differentiable with respect to the vector $\boldsymbol{x}$ of dimensions $(n \times 1)$, the Jacobian matrix (or simply *Jacobian*) of the function is defined as the $(m \times n)$ matrix

$$\boldsymbol{J}_g(\boldsymbol{x}) = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial g_1(\boldsymbol{x})}{\partial \boldsymbol{x}} \\ \dfrac{\partial g_2(\boldsymbol{x})}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial g_m(\boldsymbol{x})}{\partial \boldsymbol{x}} \end{bmatrix}. \tag{A.22}$$

If $\boldsymbol{x}(t)$ is a differentiable function with respect to t, then

$$\dot{\boldsymbol{g}}(\boldsymbol{x}) = \frac{d}{dt}\boldsymbol{g}(\boldsymbol{x}(t)) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \boldsymbol{J}_g(\boldsymbol{x})\dot{\boldsymbol{x}}. \tag{A.23}$$

A.3 Vector Operations

Given n vectors $\boldsymbol{x}_i$ of dimensions $(m \times 1)$, they are said to be *linearly independent* if the expression

$$k_1\boldsymbol{x}_1 + k_2\boldsymbol{x}_2 + \ldots + k_n\boldsymbol{x}_n = \boldsymbol{0}$$

holds true only when all the constants k_i vanish. A necessary and sufficient condition for the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_n$ to be linearly independent is that the matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n \end{bmatrix}$$

has rank n; this implies that a necessary condition for linear independence is that $n \leq m$. If instead $\varrho(\boldsymbol{A}) = r < n$, then only r vectors are linearly independent and the remaining $n - r$ vectors can be expressed as a linear combination of the previous ones.

A system of vectors $\mathcal{X}$ is a *vector space* on the field of real numbers $\mathbb{R}$ if the operations of *sum of two vectors* of $\mathcal{X}$ and *product of a scalar by a vector* of $\mathcal{X}$ have values in $\mathcal{X}$ and the following properties hold:

$$\begin{aligned}
&\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x} \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \\
&(\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} = \boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) \qquad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X} \\
&\exists \mathbf{0} \in \mathcal{X} : \boldsymbol{x} + \mathbf{0} = \boldsymbol{x} \qquad \forall \boldsymbol{x} \in \mathcal{X} \\
&\forall \boldsymbol{x} \in \mathcal{X}, \quad \exists (-\boldsymbol{x}) \in \mathcal{X} : \boldsymbol{x} + (-\boldsymbol{x}) = \mathbf{0} \\
&1\boldsymbol{x} = \boldsymbol{x} \qquad \forall \boldsymbol{x} \in \mathcal{X} \\
&\alpha(\beta\boldsymbol{x}) = (\alpha\beta)\boldsymbol{x} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x} \in \mathcal{X} \\
&(\alpha + \beta)\boldsymbol{x} = \alpha\boldsymbol{x} + \beta\boldsymbol{x} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x} \in \mathcal{X} \\
&\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha\boldsymbol{x} + \alpha\boldsymbol{y} \qquad \forall \alpha \in \mathbb{R} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}.
\end{aligned}$$

The *dimension* of the space $\dim(\mathcal{X})$ is the maximum number of linearly independent vectors $\boldsymbol{x}$ in the space. A set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ of linearly independent vectors is a *basis* of vector space $\mathcal{X}$, and each vector $\boldsymbol{y}$ in the space can be uniquely expressed as a linear combination of vectors from the basis

$$\boldsymbol{y} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \ldots + c_n\boldsymbol{x}_n, \tag{A.24}$$

where the constants $c_1, c_2, \ldots, c_n$ are said to be the *components* of the vector $\boldsymbol{y}$ in the basis $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$.

A subset $\mathcal{Y}$ of a vector space $\mathcal{X}$ is a *subspace* $\mathcal{Y} \subseteq \mathcal{X}$ if it is a vector space with the operations of vector sum and product of a scalar by a vector, i.e.,

$$\alpha\boldsymbol{x} + \beta\boldsymbol{y} \in \mathcal{Y} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Y}.$$

According to a geometric interpretation, a subspace is a *hyperplane* passing by the origin (null element) of $\mathcal{X}$.

The *scalar product* $< \boldsymbol{x}, \boldsymbol{y} >$ of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of dimensions $(m \times 1)$ is the scalar that is obtained by summing the products of the respective components in a given basis

$$< \boldsymbol{x}, \boldsymbol{y} >= x_1y_1 + x_2y_2 + \ldots + x_my_m = \boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{x}. \tag{A.25}$$

Two vectors are said to be *orthogonal* when their scalar product is null:

$$\boldsymbol{x}^T\boldsymbol{y} = 0. \tag{A.26}$$

The *norm* of a vector can be defined as

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}. \tag{A.27}$$

It is possible to show that both the *triangle inequality*

$$\|\boldsymbol{x}+\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\| \tag{A.28}$$

and the *Schwarz inequality*

$$|\boldsymbol{x}^T\boldsymbol{y}| \leq \|\boldsymbol{x}\|\,\|\boldsymbol{y}\|. \tag{A.29}$$

hold. A *unit vector* $\hat{\boldsymbol{x}}$ is a vector whose *norm* is unity, i.e., $\hat{\boldsymbol{x}}^T\hat{\boldsymbol{x}} = 1$. Given a vector $\boldsymbol{x}$, its unit vector is obtained by dividing each component by its norm:

$$\widehat{\boldsymbol{x}} = \frac{1}{\|\boldsymbol{x}\|}\boldsymbol{x}. \tag{A.30}$$

A typical example of vector space is the *Euclidean space* whose dimension is 3; in this case a basis is constituted by the unit vectors of a coordinate frame.

The *vector product* of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in the Euclidean space is the vector

$$\boldsymbol{x}\times\boldsymbol{y} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}. \tag{A.31}$$

The following properties hold:

$$\begin{aligned} &\boldsymbol{x}\times\boldsymbol{x} = \boldsymbol{0} \\ &\boldsymbol{x}\times\boldsymbol{y} = -\boldsymbol{y}\times\boldsymbol{x} \\ &\boldsymbol{x}\times(\boldsymbol{y}+\boldsymbol{z}) = \boldsymbol{x}\times\boldsymbol{y} + \boldsymbol{x}\times\boldsymbol{z}. \end{aligned}$$

The vector product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed also as the product of a matrix operator $\boldsymbol{S}(\boldsymbol{x})$ by the vector $\boldsymbol{y}$. In fact, by introducing the *skew-symmetric* matrix

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{A.32}$$

obtained with the components of vector $\boldsymbol{x}$, the vector product $\boldsymbol{x}\times\boldsymbol{y}$ is given by

$$\boldsymbol{x}\times\boldsymbol{y} = \boldsymbol{S}(\boldsymbol{x})\boldsymbol{y} = -\boldsymbol{S}(\boldsymbol{y})\boldsymbol{x} \tag{A.33}$$

as can be easily verified. Moreover, the following properties hold:

$$\begin{aligned} &\boldsymbol{S}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{S}^T(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{0} \\ &\boldsymbol{S}(\alpha\boldsymbol{x}+\beta\boldsymbol{y}) = \alpha\boldsymbol{S}(\boldsymbol{x}) + \beta\boldsymbol{S}(\boldsymbol{y}). \end{aligned}$$

Given three vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ in the Euclidean space, the following expressions hold for the *scalar triple products*:

$$\boldsymbol{x}^T(\boldsymbol{y}\times\boldsymbol{z}) = \boldsymbol{y}^T(\boldsymbol{z}\times\boldsymbol{x}) = \boldsymbol{z}^T(\boldsymbol{x}\times\boldsymbol{y}). \tag{A.34}$$

If any two vectors of three are equal, then the scalar triple product is null; e.g.,

$$\boldsymbol{x}^T(\boldsymbol{x}\times\boldsymbol{y}) = \boldsymbol{0}.$$

A.4 Linear Transformation

Consider a vector space $\mathcal{X}$ of dimension n and a vector space $\mathcal{Y}$ of dimension m with $m \leq n$. The *linear transformation* (or linear map) between the vectors $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ can be defined as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{A.35}$$

in terms of the matrix $\boldsymbol{A}$ of dimensions $(m \times n)$. The *range space* (or simply range) of the transformation is the subspace

$$\mathcal{R}(\boldsymbol{A}) = \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},\ \boldsymbol{x} \in \mathcal{X}\} \subseteq \mathcal{Y}, \tag{A.36}$$

which is the subspace generated by the linearly independent columns of matrix $\boldsymbol{A}$ taken as a basis of $\mathcal{Y}$. It is easy to recognize that

$$\varrho(\boldsymbol{A}) = \dim(\mathcal{R}(\boldsymbol{A})). \tag{A.37}$$

On the other hand, the *null space* (or simply null) of the transformation is the subspace

$$\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{x} \in \mathcal{X}\} \subseteq \mathcal{X}. \tag{A.38}$$

Given a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$, the notable result holds:

$$\varrho(\boldsymbol{A}) + \dim(\mathcal{N}(\boldsymbol{A})) = n. \tag{A.39}$$

Therefore, if $\varrho(\boldsymbol{A}) = r \leq \min\{m, n\}$, then $\dim(\mathcal{R}(\boldsymbol{A})) = r$ and $\dim(\mathcal{N}(\boldsymbol{A})) = n - r$. It follows that if $m < n$, then $\mathcal{N}(\boldsymbol{A}) \neq \emptyset$ independently of the rank of $\boldsymbol{A}$; if $m = n$, then $\mathcal{N}(\boldsymbol{A}) \neq \emptyset$ only in the case of $\varrho(\boldsymbol{A}) = r < m$.

If $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$ and $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{A}^T)$, then $\boldsymbol{y}^T\boldsymbol{x} = 0$, i.e., the vectors in the null space of $\boldsymbol{A}$ are orthogonal to each vector in the range space of the transpose of $\boldsymbol{A}$. It can be shown that the set of vectors orthogonal to each vector of the range space of $\boldsymbol{A}^T$ coincides with the null space of $\boldsymbol{A}$, whereas the set of vectors orthogonal to each vector in the null space of $\boldsymbol{A}^T$ coincides with the range space of $\boldsymbol{A}$. In symbols:

$$\mathcal{N}(\boldsymbol{A}) \equiv \mathcal{R}^{\perp}(\boldsymbol{A}^T) \qquad \mathcal{R}(\boldsymbol{A}) \equiv \mathcal{N}^{\perp}(\boldsymbol{A}^T) \tag{A.40}$$

where $\perp$ denotes the *orthogonal complement* of a subspace.

If the matrix $\boldsymbol{A}$ in (A.35) is square and idempotent, the matrix represents the *projection* of space $\mathcal{X}$ into a subspace.

A linear transformation allows the definition of the *norm* of a matrix $\boldsymbol{A}$ induced by the norm defined for a vector $\boldsymbol{x}$ as follows. In view of the property

$$\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|\ \|\boldsymbol{x}\|, \tag{A.41}$$

the norm of $\boldsymbol{A}$ can be defined as

$$\|\boldsymbol{A}\| = \sup_{\boldsymbol{x} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \tag{A.42}$$

which can also be computed as

$$\max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|.$$

A direct consequence of (A.41) is the property

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{B}\|. \tag{A.43}$$

A different norm of a matrix is the *Frobenius norm* defined as

$$\|\boldsymbol{A}\|_F = \Big(\mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})\Big)^{1/2} \tag{A.44}$$

A.5 Eigenvalues and Eigenvectors

Consider the linear transformation on a vector $\boldsymbol{u}$ established by an $(n \times n)$ square matrix $\boldsymbol{A}$. If the vector resulting from the transformation has the same direction of $\boldsymbol{u}$ (with $\boldsymbol{u} \neq \boldsymbol{0}$), then

$$\boldsymbol{A}\boldsymbol{u} = \lambda\boldsymbol{u}. \tag{A.45}$$

The equation in (A.45) can be rewritten in matrix form as

$$(\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u} = \boldsymbol{0}. \tag{A.46}$$

For the homogeneous system of equations in (A.46) to have a solution different from the trivial one $\boldsymbol{u} = \boldsymbol{0}$, it must be

$$\det(\lambda\boldsymbol{I} - \boldsymbol{A}) = 0 \tag{A.47}$$

which is termed a *characteristic equation*. Its solutions $\lambda_1, \ldots, \lambda_n$ are the *eigenvalues* of matrix $\boldsymbol{A}$; they coincide with the eigenvalues of matrix $\boldsymbol{A}^T$. On the assumption of distinct eigenvalues, the n vectors $\boldsymbol{u}_i$ satisfying the equation

$$(\lambda_i\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u}_i = \boldsymbol{0} \qquad i = 1, \ldots, n \tag{A.48}$$

are said to be the *eigenvectors* associated with the eigenvalues λ_i.

The matrix $\boldsymbol{U}$ formed by the column vectors $\boldsymbol{u}_i$ is invertible and constitutes a basis in the space of dimension n. Further, the *similarity transformation* established by $\boldsymbol{U}$

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{-1}\boldsymbol{A}\boldsymbol{U} \tag{A.49}$$

is so that $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$. It follows that $\det(\boldsymbol{A}) = \prod_{i=1}^n \lambda_i$.

If the matrix $\boldsymbol{A}$ is *symmetric*, its eigenvalues are real and $\boldsymbol{\Lambda}$ can be written as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^T\boldsymbol{A}\boldsymbol{U}; \tag{A.50}$$

hence, the eigenvector matrix $\boldsymbol{U}$ is orthogonal.

A.6 Bilinear Forms and Quadratic Forms

A *bilinear form* in the variables x_i and y_j is the scalar

$$B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j$$

which can be written in matrix form

$$B(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y} = \boldsymbol{y}^T \boldsymbol{A}^T \boldsymbol{x} \tag{A.51}$$

where $\boldsymbol{x} = [\, x_1 \quad x_2 \quad \dots \quad x_m \,]^T$, $\boldsymbol{y} = [\, y_1 \quad y_2 \quad \dots \quad y_n \,]^T$, and $\boldsymbol{A}$ is the $(m \times n)$ matrix of the coefficients a_{ij} representing the core of the form.

A special case of bilinear form is the *quadratic form*

$$Q(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \tag{A.52}$$

where $\boldsymbol{A}$ is an $(n \times n)$ square matrix. Hence, for computation of (A.52), the matrix $\boldsymbol{A}$ can be replaced with its symmetric part $\boldsymbol{A}_s$ given by (A.6). It follows that if $\boldsymbol{A}$ is a *skew-symmetric* matrix, then

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \boldsymbol{0} \qquad \forall \boldsymbol{x}.$$

The quadratic form (A.52) is said to be *positive definite* if

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0 \quad \forall \boldsymbol{x} \neq \boldsymbol{0} \qquad\qquad \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = 0 \quad \boldsymbol{x} = \boldsymbol{0}. \tag{A.53}$$

The matrix $\boldsymbol{A}$ core of the form is also said to be *positive definite*. Analogously, a quadratic form is said to be *negative definite* if it can be written as $-Q(\boldsymbol{x}) = -\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ where $Q(\boldsymbol{x})$ is positive definite.

A necessary condition for a square matrix to be positive definite is that its elements on the diagonal are strictly positive. Further, in view of (A.50), the eigenvalues of a positive definite matrix are all positive. If the eigenvalues are not known, a necessary and sufficient condition for a symmetric matrix to be positive definite is that its principal minors are strictly positive (*Sylvester criterion*). It follows that a positive definite matrix is full-rank and thus it is always invertible.

A symmetric positive definite matrix $\boldsymbol{A}$ can always be decomposed as

$$\boldsymbol{A} = \boldsymbol{U}^T \boldsymbol{\Lambda} \boldsymbol{U} \tag{A.54}$$

where $\boldsymbol{U}$ is an orthogonal matrix of eigenvectors ($\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}$) and $\boldsymbol{\Lambda}$ is the diagonal matrix of the eigenvalues of $\boldsymbol{A}$.

Let $\lambda_{\min}(\boldsymbol{A})$ and $\lambda_{\max}(\boldsymbol{A})$ respectively denote the smallest and largest eigenvalues of a positive definite matrix $\boldsymbol{A}$ ($\lambda_{\min}, \lambda_{\max} > 0$). Then, the quadratic form in (A.52) satisfies the following inequality:

$$\lambda_{\min}(\boldsymbol{A}) \|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{A}) \|\boldsymbol{x}\|^2. \tag{A.55}$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *positive semi-definite* if

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0 \qquad \forall \boldsymbol{x}. \tag{A.56}$$

This definition implies that $\varrho(\boldsymbol{A}) = r < n$, and thus r eigenvalues of $\boldsymbol{A}$ are positive and $n - r$ are null. Therefore, a positive semi-definite matrix $\boldsymbol{A}$ has a null space of finite dimension, and specifically the form vanishes when $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$. A typical example of a positive semi-definite matrix is the matrix $\boldsymbol{A} = \boldsymbol{H}^T \boldsymbol{H}$ where $\boldsymbol{H}$ is an $(m \times n)$ matrix with $m < n$. In an analogous way, a *negative semi-definite* matrix can be defined.

Given the *bilinear form* in (A.51), the *gradient* of the form with respect to $\boldsymbol{x}$ is given by

$$\nabla_{\boldsymbol{x}} B(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\partial B(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \right)^T = \boldsymbol{A} \boldsymbol{y}, \tag{A.57}$$

whereas the gradient of B with respect to $\boldsymbol{y}$ is given by

$$\nabla_{\boldsymbol{y}} B(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\partial B(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}} \right)^T = \boldsymbol{A}^T \boldsymbol{x}. \tag{A.58}$$

Given the *quadratic form* in (A.52) with $\boldsymbol{A}$ *symmetric*, the *gradient* of the form with respect to $\boldsymbol{x}$ is given by

$$\nabla_{\boldsymbol{x}} Q(\boldsymbol{x}) = \left(\frac{\partial Q(\boldsymbol{x})}{\partial \boldsymbol{x}} \right)^T = 2 \boldsymbol{A} \boldsymbol{x}. \tag{A.59}$$

Further, if $\boldsymbol{x}$ and $\boldsymbol{A}$ are differentiable functions of t, then

$$\dot{Q}(x) = \frac{d}{dt} Q(\boldsymbol{x}(t)) = 2 \boldsymbol{x}^T \boldsymbol{A} \dot{\boldsymbol{x}} + \boldsymbol{x}^T \dot{\boldsymbol{A}} \boldsymbol{x}; \tag{A.60}$$

if $\boldsymbol{A}$ is constant, then the second term obviously vanishes.

A.7 Pseudo-inverse

The inverse of a matrix can be defined only when the matrix is square and nonsingular. The inverse operation can be extended to the case of non-square matrices. Consider a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ with $\varrho(\boldsymbol{A}) = \min\{m, n\}$

If $m < n$, a *right inverse* of $\boldsymbol{A}$ can be defined as the matrix $\boldsymbol{A}_r$ of dimensions $(n \times m)$ so that

$$\boldsymbol{A} \boldsymbol{A}_r = \boldsymbol{I}_m.$$

If instead $m > n$, a *left inverse* of $\boldsymbol{A}$ can be defined as the matrix $\boldsymbol{A}_l$ of dimensions $(n \times m)$ so that

$$\boldsymbol{A}_l \boldsymbol{A} = \boldsymbol{I}_n.$$

If $\boldsymbol{A}$ has more columns than rows ($m < n$) and has rank m, a special right inverse is the matrix

$$\boldsymbol{A}_r^\dagger = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1} \tag{A.61}$$

which is termed *right pseudo-inverse*, since $\boldsymbol{A}\boldsymbol{A}_r^\dagger = \boldsymbol{I}_m$. If $\boldsymbol{W}_r$ is an $(n \times n)$ *positive definite* matrix, a *weighted* right pseudo-inverse is given by

$$\boldsymbol{A}_r^\dagger = \boldsymbol{W}_r^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{W}_r^{-1}\boldsymbol{A}^T)^{-1}. \tag{A.62}$$

If $\boldsymbol{A}$ has more rows than columns ($m > n$) and has rank n, a special left inverse is the matrix

$$\boldsymbol{A}_l^\dagger = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \tag{A.63}$$

which is termed *left pseudo-inverse*, since $\boldsymbol{A}_l^\dagger\boldsymbol{A} = \boldsymbol{I}_n$.[3] If $\boldsymbol{W}_l$ is an $(m \times m)$ *positive definite* matrix, a *weighted* left pseudo-inverse is given by

$$\boldsymbol{A}_l^\dagger = (\boldsymbol{A}^T\boldsymbol{W}_l\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{W}_l. \tag{A.64}$$

The pseudo-inverse is very useful to invert a linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with $\boldsymbol{A}$ a full-rank matrix. If $\boldsymbol{A}$ is a square nonsingular matrix, then obviously $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$ and then $\boldsymbol{A}_l^\dagger = \boldsymbol{A}_r^\dagger = \boldsymbol{A}^{-1}$.

If $\boldsymbol{A}$ has more columns than rows ($m < n$) and has rank m, then the solution $\boldsymbol{x}$ for a given $\boldsymbol{y}$ is not unique; it can be shown that the expression

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{k}, \tag{A.65}$$

with $\boldsymbol{k}$ an arbitrary $(n \times 1)$ vector and $\boldsymbol{A}^\dagger$ as in (A.61), is a solution to the system of linear equations established by (A.35). The term $\boldsymbol{A}^\dagger\boldsymbol{y} \in \mathcal{N}^\perp(\boldsymbol{A}) \equiv \mathcal{R}(\boldsymbol{A}^T)$ minimizes the norm of the solution $\|\boldsymbol{x}\|$. The term $(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{k}$ is the projection of $\boldsymbol{k}$ in $\mathcal{N}(\boldsymbol{A})$ and is termed *homogeneous solution*; as $\boldsymbol{k}$ varies, all the solutions to the homogeneous equation system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ associated with (A.35) are generated.

On the other hand, if $\boldsymbol{A}$ has more rows than columns ($m > n$), the equation in (A.35) has no solution; it can be shown that an *approximate* solution is given by

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{y} \tag{A.66}$$

where $\boldsymbol{A}^\dagger$ as in (A.63) minimizes $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|$. If instead $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{A})$, then (A.66) is a real solution.

Notice that the use of the weighted (left or right) pseudo-inverses in the solution to the linear equation systems leads to analogous results where the minimized norms are weighted according to the metrics defined by matrices $\boldsymbol{W}_r$ and $\boldsymbol{W}_l$, respectively.

The results of this section can be easily extended to the case of (square or nonsquare) matrices $\boldsymbol{A}$ not having full-rank. In particular, the expression (A.66) (with the pseudo-inverse computed by means of the singular value decomposition of $\boldsymbol{A}$) gives the minimum-norm vector among all those minimizing $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|$.

[3] Subscripts l and r are usually omitted whenever the use of a left or right pseudo-inverse is clear from the context.

A.8 Singular Value Decomposition

For a nonsquare matrix it is not possible to define eigenvalues. An extension of the eigenvalue concept can be obtained by singular values. Given a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$, the matrix $\boldsymbol{A}^T\boldsymbol{A}$ has n nonnegative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ (ordered from the largest to the smallest) which can be expressed in the form

$$\lambda_i = \sigma_i^2 \qquad \sigma_i \geq 0.$$

The scalars $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are said to be the *singular values* of matrix $\boldsymbol{A}$. The *singular value decomposition* (SVD) of matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{A.67}$$

where $\boldsymbol{U}$ is an $(m \times m)$ orthogonal matrix

$$\boldsymbol{U} = [\,\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \ldots \quad \boldsymbol{u}_m\,], \tag{A.68}$$

$\boldsymbol{V}$ is an $(n \times n)$ orthogonal matrix

$$\boldsymbol{V} = [\,\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \ldots \quad \boldsymbol{v}_n\,] \tag{A.69}$$

and $\boldsymbol{\Sigma}$ is an $(m \times n)$ matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{D} = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\} \tag{A.70}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. The number of non-null singular values is equal to the rank r of matrix $\boldsymbol{A}$.

The columns of $\boldsymbol{U}$ are the eigenvectors of the matrix $\boldsymbol{A}\boldsymbol{A}^T$, whereas the columns of $\boldsymbol{V}$ are the eigenvectors of the matrix $\boldsymbol{A}^T\boldsymbol{A}$. In view of the partitions of $\boldsymbol{U}$ and $\boldsymbol{V}$ in (A.68), (A.69), it is $\boldsymbol{A}\boldsymbol{v}_i = \sigma_i\boldsymbol{u}_i$, for $i = 1, \ldots, r$ and $\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{0}$, for $i = r+1, \ldots, n$.

Singular value decomposition is useful for analysis of the linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ established in (A.35). According to a geometric interpretation, the matrix $\boldsymbol{A}$ transforms the unit sphere in IR^n defined by $\|\boldsymbol{x}\| = 1$ into the set of vectors $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ which define an *ellipsoid* of dimension r in IR^m. The singular values are the lengths of the various axes of the ellipsoid. The *condition number* of the matrix

$$\kappa = \frac{\sigma_1}{\sigma_r}$$

is related to the eccentricity of the ellipsoid and provides a measure of ill-conditioning ($\kappa \gg 1$) for numerical solution of the system established by (A.35).

It is worth noticing that the numerical procedure of singular value decomposition is commonly adopted to compute the (right or left) pseudo-inverse $\boldsymbol{A}^\dagger$, even in the case of a matrix $\boldsymbol{A}$ not having full rank. In fact, from (A.67), (A.70) it is

$$\boldsymbol{A}^\dagger = \boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T \tag{A.71}$$

with

$$\boldsymbol{\Sigma}^\dagger = \begin{bmatrix} \boldsymbol{D}^\dagger & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{D}^\dagger = \operatorname{diag}\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}\right\}. \tag{A.72}$$

Bibliography

A reference text on linear algebra is [169]. For matrix computation see [88]. The properties of pseudo-inverse matrices are discussed in [24].

B

Rigid-body Mechanics

The goal of this appendix is to recall some fundamental concepts of *rigid body mechanics* which are preliminary to the study of manipulator *kinematics*, *statics* and *dynamics*.

B.1 Kinematics

A *rigid body* is a system characterized by the constraint that the distance between any two points is always constant.

Consider a rigid body $\mathcal{B}$ moving with respect to an orthonormal reference frame O–xyz of unit vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, called *fixed frame*. The rigidity assumption allows the introduction of an orthonormal frame O'–$x'y'z'$ attached to the body, called *moving frame*, with respect to which the position of any point of $\mathcal{B}$ is independent of time. Let $\boldsymbol{x}'(t)$, $\boldsymbol{y}'(t)$, $\boldsymbol{z}'(t)$ be the unit vectors of the moving frame expressed in the fixed frame at time t.

The orientation of the moving frame O'–$x'y'z'$ at time t with respect to the fixed frame O–xyz can be expressed by means of the *orthogonal* (3×3) matrix

$$\boldsymbol{R}(t) = \begin{bmatrix} \boldsymbol{x}'^T(t)\boldsymbol{x} & \boldsymbol{y}'^T(t)\boldsymbol{x} & \boldsymbol{z}'^T(t)\boldsymbol{x} \\ \boldsymbol{x}'^T(t)\boldsymbol{y} & \boldsymbol{y}'^T(t)\boldsymbol{y} & \boldsymbol{z}'^T(t)\boldsymbol{y} \\ \boldsymbol{x}'^T(t)\boldsymbol{z} & \boldsymbol{y}'^T(t)\boldsymbol{z} & \boldsymbol{z}'^T(t)\boldsymbol{z} \end{bmatrix}, \tag{B.1}$$

which is termed *rotation matrix* defined in the orthonormal special group $SO(3)$ of the (3×3) matrices with orthonormal columns and determinant equal to 1. The columns of the matrix in (B.1) represent the components of the unit vectors of the moving frame when expressed in the fixed frame, whereas the rows represent the components of the unit vectors of the fixed frame when expressed in the moving frame.

Let $\boldsymbol{p}'$ be the *constant* position vector of a generic point P of $\mathcal{B}$ in the moving frame O'–$x'y'z'$. The motion of P with respect to the fixed frame O–xyz is described by the equation

$$\boldsymbol{p}(t) = \boldsymbol{p}_{O'}(t) + \boldsymbol{R}(t)\boldsymbol{p}', \tag{B.2}$$

where $\boldsymbol{p}_{O'}(t)$ is the position vector of origin O' of the moving frame with respect to the fixed frame.

Notice that a position vector is a *bound vector* since its line of application and point of application are both prescribed, in addition to its direction; the point of application typically coincides with the origin of a reference frame. Therefore, to transform a bound vector from a frame to another, both translation and rotation between the two frames must be taken into account.

If the positions of the points of $\mathcal{B}$ in the moving frame are known, it follows from (B.2) that the motion of each point of $\mathcal{B}$ with respect to the fixed frame is uniquely determined once the position of the origin and the orientation of the moving frame with respect to the fixed frame are specified in time. The origin of the moving frame is determined by *three* scalar functions of time. Since the orthonormality conditions impose six constraints on the nine elements of matrix $\boldsymbol{R}(t)$, the *orientation* of the moving frame depends only on *three* independent scalar functions, three being the minimum number of parameters to represent $SO(3)$.[1]

Therefore, a rigid body motion is described by arbitrarily specifying *six* scalar functions of time, which describe the body *pose* (position + orientation). The resulting rigid motions belong to the *special Euclidean group* $SE(3) = \mathbb{R}^3 \times SO(3)$.

The expression in (B.2) continues to hold if the position vector $\boldsymbol{p}_{O'}(t)$ of the origin of the moving frame is replaced with the position vector of any other point of $\mathcal{B}$, i.e.,

$$\boldsymbol{p}(t) = \boldsymbol{p}_Q(t) + \boldsymbol{R}(t)(\boldsymbol{p}' - \boldsymbol{p}'_Q) \qquad \text{(B.3)}$$

where $\boldsymbol{p}_Q(t)$ and $\boldsymbol{p}'_Q$ are the position vectors of a point Q of $\mathcal{B}$ in the fixed and moving frames, respectively.

In the following, for simplicity of notation, the dependence on the time variable t will be dropped.

Differentiating (B.3) with respect to time gives the known velocity composition rule

$$\dot{\boldsymbol{p}} = \dot{\boldsymbol{p}}_Q + \boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q), \qquad \text{(B.4)}$$

where $\boldsymbol{\omega}$ is the *angular velocity* of rigid body $\mathcal{B}$. Notice that $\boldsymbol{\omega}$ is a *free vector* since its point of application is not prescribed. To transform a free vector from a frame to another, only rotation between the two frames must be taken into account.

By recalling the definition of the skew-symmetric operator $\boldsymbol{S}(\cdot)$ in (A.32), the expression in (B.4) can be rewritten as

$$\begin{aligned}\dot{\boldsymbol{p}} &= \dot{\boldsymbol{p}}_Q + \boldsymbol{S}(\boldsymbol{\omega})(\boldsymbol{p} - \boldsymbol{p}_Q)\\ &= \dot{\boldsymbol{p}}_Q + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}(\boldsymbol{p}' - \boldsymbol{p}'_Q).\end{aligned}$$

[1] The minimum number of parameters represent a special orthonormal group $SO(m)$ is equal to $m(m-1)/2$.

Comparing this equation with the formal time derivative of (B.3) leads to the result

$$\dot{\boldsymbol{R}} = \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}. \tag{B.5}$$

In view of (B.4), the *elementary displacement* of a point P of the rigid body $\mathcal{B}$ in the time interval $(t, t+dt)$ is

$$\begin{aligned} d\boldsymbol{p} = \dot{\boldsymbol{p}}dt &= \big(\dot{\boldsymbol{p}}_Q + \boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q)\big)dt \\ &= d\boldsymbol{p}_Q + \boldsymbol{\omega}dt \times (\boldsymbol{p} - \boldsymbol{p}_Q). \end{aligned} \tag{B.6}$$

Differentiating (B.4) with respect to time yields the following expression for acceleration:

$$\ddot{\boldsymbol{p}} = \ddot{\boldsymbol{p}}_Q + \dot{\boldsymbol{\omega}} \times (\boldsymbol{p} - \boldsymbol{p}_Q) + \boldsymbol{\omega} \times \big(\boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q)\big). \tag{B.7}$$

B.2 Dynamics

Let ρdV be the mass of an elementary particle of a rigid body $\mathcal{B}$, where ρ denotes the density of the particle of volume dV. Also let $V_{\mathcal{B}}$ be the body volume and $m = \int_{V_{\mathcal{B}}} \rho dV$ its *total mass* assumed to be constant. If $\boldsymbol{p}$ denotes the position vector of the particle of mass ρdV in the frame O–xyz, the *centre of mass* of $\mathcal{B}$ is defined as the point C whose position vector is

$$\boldsymbol{p}_C = \frac{1}{m}\int_{V_{\mathcal{B}}} \boldsymbol{p}\rho dV. \tag{B.8}$$

In the case when $\mathcal{B}$ is the union of n distinct parts of mass $m_1, \ldots, m_n$ and centres of mass $\boldsymbol{p}_{C1} \ldots \boldsymbol{p}_{Cn}$, the centre of mass of $\mathcal{B}$ can be computed as

$$\boldsymbol{p}_C = \frac{1}{m}\sum_{i=1}^{n} m_i \boldsymbol{p}_{Ci}$$

with $m = \sum_{i=1}^{n} m_i$.

Let r be a line passing by O and $d(\boldsymbol{p})$ the distance from r of the particle of $\mathcal{B}$ of mass ρdV and position vector $\boldsymbol{p}$. The *moment of inertia* of body $\mathcal{B}$ with respect to line r is defined as the positive scalar

$$I_r = \int_{V_{\mathcal{B}}} d^2(\boldsymbol{p})\rho dV.$$

Let $\boldsymbol{r}$ denote the unit vector of line r; then, the moment of inertia of $\mathcal{B}$ with respect to line r can be expressed as

$$I_r = \boldsymbol{r}^T \left(\int_{V_{\mathcal{B}}} \boldsymbol{S}^T(\boldsymbol{p})\boldsymbol{S}(\boldsymbol{p})\rho dV \right) \boldsymbol{r} = \boldsymbol{r}^T \boldsymbol{I}_O \boldsymbol{r}, \tag{B.9}$$

where $\boldsymbol{S}(\cdot)$ is the skew-symmetric operator in (A.31), and the *symmetric*, *positive definite* matrix

$$\boldsymbol{I}_O = \begin{bmatrix} \int_{V_{\mathcal{B}}}(p_y^2 + p_z^2)\rho dV & -\int_{V_{\mathcal{B}}} p_x p_y \rho dV & -\int_{V_{\mathcal{B}}} p_x p_z \rho dV \\ * & \int_{V_{\mathcal{B}}}(p_x^2 + p_z^2)\rho dV & -\int_{V_{\mathcal{B}}} p_y p_z \rho dV \\ * & * & \int_{V_{\mathcal{B}}}(p_x^2 + p_y^2)\rho dV \end{bmatrix}$$

$$= \begin{bmatrix} I_{Oxx} & -I_{Oxy} & -I_{Oxz} \\ * & I_{Oyy} & -I_{Oyz} \\ * & * & I_{Ozz} \end{bmatrix} \tag{B.10}$$

is termed *inertia tensor* of body $\mathcal{B}$ relative to pole O.[2] The (positive) elements I_{Oxx}, I_{Oyy}, I_{Ozz} are the *inertia moments* with respect to three coordinate axes of the reference frame, whereas the elements I_{Oxy}, I_{Oxz}, I_{Oyz} (of any sign) are said to be *products of inertia*.

The expression of the inertia tensor of a rigid body $\mathcal{B}$ depends both on the pole and the reference frame. If orientation of the reference frame with origin at O is changed according to a rotation matrix $\boldsymbol{R}$, the inertia tensor $\boldsymbol{I}'_O$ in the new frame is related to $\boldsymbol{I}_O$ by the relationship

$$\boldsymbol{I}_O = \boldsymbol{R}\boldsymbol{I}'_O\boldsymbol{R}^T. \tag{B.11}$$

The way an inertia tensor is transformed when the pole is changed can be inferred by the following equation, also known as *Steiner theorem* or parallel axis theorem:

$$\boldsymbol{I}_O = \boldsymbol{I}_C + m\boldsymbol{S}^T(\boldsymbol{p}_C)\boldsymbol{S}(\boldsymbol{p}_C), \tag{B.12}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass of $\mathcal{B}$, when expressed in a frame parallel to the frame with origin at O and with origin at the centre of mass C.

Since the inertia tensor is a symmetric positive definite matrix, there always exists a reference frame in which the inertia tensor attains a diagonal form; such a frame is said to be a *principal frame* (relative to pole O) and its coordinate axes are said to be *principal axes*. In the case when pole O coincides with the centre of mass, the frame is said to be a *central frame* and its axes are said to be *central axes*.

Notice that if the rigid body is moving with respect to the reference frame with origin at O, then the elements of the inertia tensor $\boldsymbol{I}_O$ become a function of time. With respect to a pole and a reference frame attached to the body (moving frame), instead, the elements of the inertia tensor represent six structural constants of the body which are known once the pole and reference frame have been specified.

[2] The symbol '$*$' has been used to avoid rewriting the symmetric elements.

Let $\dot{\boldsymbol{p}}$ be the velocity of a particle of $\mathcal{B}$ of elementary mass ρdV in frame O–xyz. The *linear momentum* of body $\mathcal{B}$ is defined as the vector

$$\boldsymbol{l} = \int_{V_{\mathcal{B}}} \dot{\boldsymbol{p}} \rho dV = m \dot{\boldsymbol{p}}_C. \tag{B.13}$$

Let Ω be any point in space and $\boldsymbol{p}_\Omega$ its position vector in frame O–xyz; then, the *angular momentum* of body $\mathcal{B}$ relative to pole Ω is defined as the vector

$$\boldsymbol{k}_\Omega = \int_{V_{\mathcal{B}}} \dot{\boldsymbol{p}} \times (\boldsymbol{p}_\Omega - \boldsymbol{p}) \rho dV.$$

The pole can be either fixed or moving with respect to the reference frame. The angular momentum of a rigid body has the following notable expression:

$$\boldsymbol{k}_\Omega = \boldsymbol{I}_C \boldsymbol{\omega} + m \dot{\boldsymbol{p}}_C \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_C), \tag{B.14}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass, when expressed in a frame parallel to the reference frame with origin at the centre of mass.

The *forces* acting on a generic system of material particles can be distinguished into *internal* forces and *external* forces.

The internal forces, exerted by one part of the system on another, have null linear and angular momentum and thus they do not influence rigid body motion.

The external forces, exerted on the system by an agency outside the system, in the case of a rigid body $\mathcal{B}$ are distinguished into *active* forces and *reaction* forces.

The active forces can be either *concentrated* forces or *body* forces. The former are applied to specific points of $\mathcal{B}$, whereas the latter act on all elementary particles of the body. An example of body force is the *gravitational force* which, for any elementary particle of mass ρdV, is equal to $\boldsymbol{g}_0 \rho dV$ where $\boldsymbol{g}_0$ is the gravity acceleration vector.

The reaction forces are those exerted because of surface contact between two or more bodies. Such forces can be distributed on the contact surfaces or they can be assumed to be concentrated.

For a rigid body $\mathcal{B}$ subject to gravitational force, as well as to active and or reaction forces $\boldsymbol{f}_1 \ldots \boldsymbol{f}_n$ concentrated at points $\boldsymbol{p}_1 \ldots \boldsymbol{p}_n$, the *resultant* of the external forces $\boldsymbol{f}$ and the *resultant moment* $\boldsymbol{\mu}_\Omega$ with respect to a pole Ω are respectively

$$\boldsymbol{f} = \int_{V_{\mathcal{B}}} \boldsymbol{g}_0 \rho dV + \sum_{i=1}^{n} \boldsymbol{f}_i = m \boldsymbol{g}_0 + \sum_{i=1}^{n} \boldsymbol{f}_i \tag{B.15}$$

$$\begin{aligned} \boldsymbol{\mu}_\Omega &= \int_{V_{\mathcal{B}}} \boldsymbol{g}_0 \times (\boldsymbol{p}_\Omega - \boldsymbol{p}) \rho dV + \sum_{i=1}^{n} \boldsymbol{f}_i \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_i) \\ &= m \boldsymbol{g}_0 \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_C) + \sum_{i=1}^{n} \boldsymbol{f}_i \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_i). \end{aligned} \tag{B.16}$$

In the case when $\boldsymbol{f}$ and $\boldsymbol{\mu}_\Omega$ are known and it is desired to compute the resultant moment with respect to a point Ω' other than Ω, the following relation holds:

$$\boldsymbol{\mu}_{\Omega'} = \boldsymbol{\mu}_\Omega + \boldsymbol{f} \times (\boldsymbol{p}_{\Omega'} - \boldsymbol{p}_\Omega). \tag{B.17}$$

Consider now a generic system of material particles subject to *external forces* of resultant $\boldsymbol{f}$ and resultant moment $\boldsymbol{\mu}_\Omega$. The motion of the system in a frame O–xyz is established by the following *fundamental principles of dynamics* (Newton laws of motion):

$$\boldsymbol{f} = \dot{\boldsymbol{l}} \tag{B.18}$$

$$\boldsymbol{\mu}_\Omega = \dot{\boldsymbol{k}}_\Omega \tag{B.19}$$

where Ω is a pole fixed or coincident with the centre of mass C of the system. These equations hold for any mechanical system and can be used even in the case of variable mass. For a system with constant mass, computing the time derivative of the momentum in (B.18) gives *Newton equations of motion* in the form

$$\boldsymbol{f} = m\ddot{\boldsymbol{p}}_C, \tag{B.20}$$

where the quantity on the right-hand side represents the *resultant of inertia forces*.

If, besides the assumption of constant mass, the assumption of rigid system holds too, the expression in (B.14) of the angular momentum with (B.19) yield *Euler equations of motion* in the form

$$\boldsymbol{\mu}_\Omega = \boldsymbol{I}_\Omega\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}_\Omega\boldsymbol{\omega}), \tag{B.21}$$

where the quantity on the right-hand side represents the *resultant moment of inertia forces*.

For a system constituted by a set of rigid bodies, the external forces obviously do not include the reaction forces exerted between the bodies belonging to the same system.

B.3 Work and Energy

Given a force $\boldsymbol{f}_i$ applied at a point of position $\boldsymbol{p}_i$ with respect to frame O–xyz, the *elementary work* of the force $\boldsymbol{f}_i$ on the displacement $d\boldsymbol{p}_i = \dot{\boldsymbol{p}}_i dt$ is defined as the scalar

$$dW_i = \boldsymbol{f}_i^T d\boldsymbol{p}_i.$$

For a rigid body $\mathcal{B}$ subject to a system of forces of resultant $\boldsymbol{f}$ and resultant moment $\boldsymbol{\mu}_Q$ with respect to any point Q of $\mathcal{B}$, the elementary work on the rigid displacement (B.6) is given by

$$dW = (\boldsymbol{f}^T\dot{\boldsymbol{p}}_Q + \boldsymbol{\mu}_Q^T\boldsymbol{\omega})dt = \boldsymbol{f}^T d\boldsymbol{p}_Q + \boldsymbol{\mu}_Q^T\boldsymbol{\omega}dt. \tag{B.22}$$

The *kinetic energy* of a body $\mathcal{B}$ is defined as the scalar quantity

$$\mathcal{T} = \frac{1}{2}\int_{V_{\mathcal{B}}} \dot{\boldsymbol{p}}^T \dot{\boldsymbol{p}} \rho dV$$

which, for a rigid body, takes on the notable expression

$$\mathcal{T} = \frac{1}{2} m \dot{\boldsymbol{p}}_C^T \dot{\boldsymbol{p}}_C + \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{I}_C \boldsymbol{\omega} \tag{B.23}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass expressed in a frame parallel to the reference frame with origin at the centre of mass.

A system of position forces, i.e., the forces depending only on the positions of the points of application, is said to be *conservative* if the work done by each force is independent of the trajectory described by the point of application of the force but it depends only on the initial and final positions of the point of application. In this case, the elementary work of the system of forces is equal to minus the total differential of a scalar function termed *potential energy*, i.e.,

$$dW = -d\mathcal{U}. \tag{B.24}$$

An example of a conservative system of forces on a rigid body is the gravitational force, with which is associated the potential energy

$$\mathcal{U} = -\int_{V_{\mathcal{B}}} \boldsymbol{g}_0^T \boldsymbol{p} \rho dV = -m \boldsymbol{g}_0^T \boldsymbol{p}_C. \tag{B.25}$$

B.4 Constrained Systems

Consider a system $\mathcal{B}_r$ of r rigid bodies and assume that all the elements of $\mathcal{B}_r$ can reach any position in space. In order to find uniquely the position of all the points of the system, it is necessary to assign a vector $\boldsymbol{x} = [\,x_1 \quad \dots \quad x_p\,]^T$ of $6r = p$ parameters, termed *configuration*. These parameters are termed *Lagrange* or *generalized coordinates* of the *unconstrained* system $\mathcal{B}_r$, and p determines the number of *degrees of freedom* (DOFs).

Any limitation on the mobility of the system $\mathcal{B}_r$ is termed *constraint*. A constraint acting on $\mathcal{B}_r$ is said to be *holonomic* if it is expressed by a system of equations

$$\boldsymbol{h}(\boldsymbol{x}, t) = \boldsymbol{0}, \tag{B.26}$$

where $\boldsymbol{h}$ is a vector of dimensions $(s \times 1)$, with $s < m$. On the other hand, a constraint in the form $\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) = \boldsymbol{0}$ which is nonintegrable is said to be *nonholonomic*. For simplicity, only equality (or *bilateral*) constraints are considered. If the equations in (B.26) do not explicitly depend on time, the constraint is said to be *scleronomic*.

On the assumption that $\boldsymbol{h}$ has continuous and continuously differentiable components, and its Jacobian $\partial \boldsymbol{h}/\partial \boldsymbol{x}$ has full rank, the equations in (B.26) allow the elimination of s out of m coordinates of the system $\mathcal{B}_r$. With the remaining $n = m - s$ coordinates it is possible to determine uniquely the configurations of $\mathcal{B}_r$ satisfying the constraints (B.26). Such coordinates are the *Lagrange* or *generalized coordinates* and n is the number of *degrees of freedom* of the *unconstrained* system $\mathcal{B}_r$.[3]

The motion of a system $\mathcal{B}_r$ with n DOFs and holonomic equality constraints can be described by equations of the form

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{q}(t), t), \tag{B.27}$$

where $\boldsymbol{q}(t) = [\, q_1(t) \quad \dots \quad q_n(t)\,]^T$ is a vector of Lagrange coordinates.

The *elementary displacement* of system (B.27) relative to the interval $(t, t+dt)$ is defined as

$$d\boldsymbol{x} = \frac{\partial \boldsymbol{x}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}}dt + \frac{\partial \boldsymbol{x}(\boldsymbol{q},t)}{\partial t}dt. \tag{B.28}$$

The *virtual displacement* of system (B.27) at time t, relative to an increment $\delta\boldsymbol{\lambda}$, is defined as the quantity

$$\delta\boldsymbol{x} = \frac{\partial \boldsymbol{x}(\boldsymbol{q},t)}{\partial \boldsymbol{q}}\delta\boldsymbol{q}. \tag{B.29}$$

The difference between the elementary displacement and the virtual displacement is that the former is relative to an actual motion of the system in an interval $(t, t+dt)$ which is consistent with the constraints, while the latter is relative to an imaginary motion of the system when the constraints are made invariant and equal to those at time t.

For a system with time-invariant constraints, the equations of motion (B.27) become

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{q}(t)), \tag{B.30}$$

and then, by setting $\delta\boldsymbol{\lambda} = d\boldsymbol{\lambda} = \dot{\boldsymbol{\lambda}}dt$, the virtual displacements (B.29) coincide with the elementary displacements (B.28).

To the concept of virtual displacement can be associated that of *virtual work* of a system of forces, by considering a virtual displacement instead of an elementary displacement.

If external forces are distinguished into *active forces* and *reaction forces*, a direct consequence of the principles of dynamics (B.18), (B.19) applied to the system of rigid bodies $\mathcal{B}_r$ is that, for each virtual displacement, the following relation holds:

$$\delta W_m + \delta W_a + \delta W_h = 0, \tag{B.31}$$

where δW_m, δW_a, δW_h are the total virtual works done by the inertia, active, reaction forces, respectively.

[3] In general, the Lagrange coordinates of a constrained system have a local validity; in certain cases, such as the joint variables of a manipulator, they can have a global validity.

In the case of *frictionless* equality constraints, reaction forces are exerted orthogonally to the contact surfaces and the virtual work is always null. Hence, (B.31) reduces to

$$\delta W_m + \delta W_a = 0. \tag{B.32}$$

For a steady system, inertia forces are identically null. Then the condition for the equilibrium of system $\mathcal{B}_r$ is that the virtual work of the active forces is identically null on any virtual displacement, which gives the fundamental equation of *statics* of a constrained system

$$\delta W_a = 0 \tag{B.33}$$

known as *principle of virtual work*. Expressing (B.33) in terms of the increment $\delta\boldsymbol{\lambda}$ of generalized coordinates leads to

$$\delta W_a = \boldsymbol{\zeta}^T \delta\boldsymbol{q} = 0 \tag{B.34}$$

where $\boldsymbol{\zeta}$ denotes the $(n \times 1)$ vector of active *generalized* forces.

In the dynamic case, it is worth distinguishing active forces into *conservative* (that can be derived from a potential) and *nonconservative*. The virtual work of conservative forces is given by

$$\delta W_c = -\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}} \delta\boldsymbol{q}, \tag{B.35}$$

where $\mathcal{U}(\boldsymbol{\lambda})$ is the total potential energy of the system. The work of nonconservative forces can be expressed in the form

$$\delta W_{nc} = \boldsymbol{\xi}^T \delta\boldsymbol{q}, \tag{B.36}$$

where $\boldsymbol{\xi}$ denotes the vector of nonconservative generalized forces. It follows that the vector of active generalized forces is

$$\boldsymbol{\zeta} = \boldsymbol{\xi} - \left(\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}}\right)^T. \tag{B.37}$$

Moreover, the work of inertia forces can be computed from the total kinetic energy of system $\mathcal{T}$ as

$$\delta W_m = \left(\frac{\partial \mathcal{T}}{\partial \boldsymbol{q}} - \frac{d}{dt}\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{q}}}\right) \delta\boldsymbol{q}. \tag{B.38}$$

Substituting (B.35), (B.36), (B.38) into (B.32) and observing that (B.32) holds true for any increment $\delta\boldsymbol{\lambda}$ leads to *Lagrange equations*

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^T = \boldsymbol{\xi}, \tag{B.39}$$

where

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{B.40}$$

is the *Lagrangian* function of the system. The equations in (B.39) completely describe the dynamic behaviour of an n-DOF system with holonomic equality constraints.

The sum of kinetic and potential energy of a system with time-invariant constraints is termed *Hamiltonian* function

$$\mathcal{H} = \mathcal{T} + \mathcal{U}. \tag{B.41}$$

Conservation of energy dictates that the time derivative of the Hamiltonian must balance the power generated by the nonconservative forces acting on the system, i.e.,

$$\frac{d\mathcal{H}}{dt} = \boldsymbol{\xi}^T \dot{\boldsymbol{q}}. \tag{B.42}$$

In view of (B.37), (B.41), the equation in (B.42) becomes

$$\frac{d\mathcal{T}}{dt} = \boldsymbol{\zeta}^T \dot{\boldsymbol{q}}. \tag{B.43}$$

Bibliography

The fundamental concepts of rigid-body mechanics and constrained systems can be found in classical texts such as [87, 154, 224]. An authoritative reference on rigid-body system dynamics is [187].

C

Feedback Control

As a premise to the study of manipulator decentralized control and centralized control, the fundamental principles of *feedback control* of *linear systems* are recalled, and an approach to the determination of control laws for *nonlinear systems* based on the use of *Lyapunov functions* is presented.

C.1 Control of Single-input/Single-output Linear Systems

According to classical *automatic control* theory of *linear time-invariant single-input/single-output systems*, in order to servo the output $y(t)$ of a system to a reference $r(t)$, it is worth adopting a *negative feedback control* structure. This structure indeed allows the use of approximate mathematical models to describe the input/output relationship of the system to control, since negative feedback has a potential for reducing the effects of system parameter variations and nonmeasurable disturbance inputs $d(t)$ on the output.

This structure can be represented in the *domain of complex variable s* as in the block scheme of Fig. C.1, where $G(s)$, $H(s)$ and $C(s)$ are the transfer functions of the system to control, the transducer and the controller, respectively. From this scheme it is easy to derive

$$Y(s) = W(s)R(s) + W_D(s)D(s), \tag{C.1}$$

where

$$W(s) = \frac{C(s)G(s)}{1 + C(s)G(s)H(s)} \tag{C.2}$$

is the *closed-loop input/output transfer function* and

$$W_D(s) = \frac{G(s)}{1 + C(s)G(s)H(s)} \tag{C.3}$$

is the *disturbance/output transfer function*.

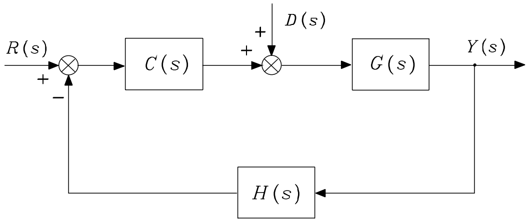

Fig. C.1. Feedback control structure

The goal of the controller design is to find a control structure $C(s)$ ensuring that the output variable $Y(s)$ tracks a reference input $R(s)$. Further, the controller should guarantee that the effects of the disturbance input $D(s)$ on the output variable are suitably reduced. The goal is then twofold, namely, *reference tracking* and *disturbance rejection*.

The basic problem for controller design consists of the determination of an action $C(s)$ which can make the system *asymptotically stable*. In the absence of positive or null real part pole/zero and zero/pole cancellation in the *open-loop* function $F(s) = C(s)G(s)H(s)$, a necessary and sufficient condition for asymptotic stability is that the *poles* of $W(s)$ and $W_D(s)$ have all *negative real parts*; such poles coincide with the zeros of the rational transfer function $1 + F(s)$. Testing for this condition can be performed by resorting to stability criteria, thus avoiding computation of the function zeros.

Routh criterion allows the determination of the sign of the real parts of the zeros of the function $1 + F(s)$ by constructing a table with the coefficients of the polynomial at the numerator of $1 + F(s)$ (*characteristic polynomial*).

Routh criterion is easy to apply for testing stability of a feedback system, but it does not provide a direct relationship between the open-loop function and stability of the closed-loop system. It is then worth resorting to *Nyquist criterion* which is based on the representation, in the complex plane, of the open-loop transfer function $F(s)$ evaluated in the *domain of real angular frequency* ($s = j\omega, -\infty < \omega < +\infty$).

Drawing of Nyquist plot and computation of the number of circles made by the vector representing the complex number $1 + F(j\omega)$ when ω continuously varies from $-\infty$ to $+\infty$ allows a test on whether or not the *closed-loop* system is asymptotically stable. It is also possible to determine the number of positive, null and negative real part roots of the characteristic polynomial, similarly to application of Routh criterion. Nonetheless, Nyquist criterion is based on the plot of the open-loop transfer function, and thus it allows the determination of a direct relationship between this function and closed-loop system stability. It is then possible from an examination of the Nyquist plot to draw suggestions on the controller structure $C(s)$ which ensures closed-loop system asymptotic stability.

If the closed-loop system is asymptotically stable, the *steady-state response* to a sinusoidal input $r(t)$, with $d(t) = 0$, is sinusoidal, too. In this case, the function $W(s)$, evaluated for $s = j\omega$, is termed *frequency response function*; the frequency response function of a feedback system can be assimilated to that of a low-pass filter with the possible occurrence of a *resonance peak* inside its *bandwidth*.

As regards the transducer, this should be chosen so that its bandwidth is much greater than the feedback system bandwidth, in order to ensure a nearly instantaneous response for any value of ω inside the bandwidth of $W(j\omega)$. Therefore, setting $H(j\omega) \approx H_0$ and assuming that the *loop gain* $|C(j\omega)G(j\omega)H_0| \gg 1$ in the same bandwidth, the expression in (C.1) for $s = j\omega$ can be approximated as

$$Y(j\omega) \approx \frac{R(j\omega)}{H_0} + \frac{D(j\omega)}{C(j\omega)H_0}.$$

Assuming $R(j\omega) = H_0 Y_d(j\omega)$ leads to

$$Y(j\omega) \approx Y_d(j\omega) + \frac{D(j\omega)}{C(j\omega)H_0}; \tag{C.4}$$

i.e., the output tracks the desired output $Y_d(j\omega)$ and the frequency components of the disturbance in the bandwidth of $W(j\omega)$ produce an effect on the output which can be reduced by increasing $|C(j\omega)H_0|$. Furthermore, if the disturbance input is a constant, the steady-state output is not influenced by the disturbance as long as $C(s)$ has at least a pole at the origin.

Therefore, a feedback control system is capable of establishing a proportional relationship between the desired output and the actual output, as evidenced by (C.4). This equation, however, requires that the frequency content of the input (desired output) be inside the frequency range for which the loop gain is much greater than unity.

The previous considerations show the advantage of including a *proportional action* and an *integral action* in the controller $C(s)$, leading to the transfer function

$$C(s) = K_I \frac{1 + sT_I}{s} \tag{C.5}$$

of a *proportional-integral controller* (PI); T_I is the time constant of the integral action and the quantity $K_I T_I$ is called proportional sensitivity.

The adoption of a PI controller is effective for low-frequency response of the system, but it may involve a reduction of *stability margins* and/or a reduction of closed-loop system bandwidth. To avoid these drawbacks, a *derivative action* can be added to the proportional and integral actions, leading to the transfer function

$$C(s) = K_I \frac{1 + sT_I + s^2 T_D T_I}{s} \tag{C.6}$$

of a *proportional-integral-derivative controller* (PID); T_D denotes the time constant of the derivative action. Notice that physical realizability of (C.6) demands the introduction of a high-frequency pole which little influences the input/output relationship in the system bandwidth. The transfer function in (C.6) is characterized by the presence of two zeros which provide a stabilizing action and an enlargement of the closed-loop system bandwidth. Bandwidth enlargement implies shorter *response time* of the system, in terms of both variations of the reference signal and recovery action of the feedback system to output variations induced by the disturbance input.

The parameters of the adopted control structure should be chosen so as to satisfy requirements on the system behaviour at *steady state* and during the *transient*. Classical tools to determine such parameters are the *root locus* in the domain of the complex variable s or the *Nichols chart* in the domain of the real angular frequency ω. The two tools are conceptually equivalent. Their potential is different in that root locus allows a control law to be found which assigns the exact parameters of the closed-loop system time response, whereas Nichols chart allows a controller to be specified which confers good transient and steady-state behaviour to the system response.

A feedback system with strict requirements on the steady-state and transient behaviour, typically, has a response that can be assimilated to that of a *second-order system*. In fact, even for closed-loop functions of greater order, it is possible to identify a pair of complex conjugate poles whose real part absolute value is smaller than the real part absolute values of the other poles. Such a pair of poles is *dominant* in that its contribution to the transient response prevails over that of the other poles. It is then possible to approximate the input/output relationship with the transfer function

$$W(s) = \frac{k_W}{1 + \dfrac{2\zeta s}{\omega_n} + \dfrac{s^2}{\omega_n^2}} \tag{C.7}$$

which has to be realized by a proper choice of the controller. Regarding the values to assign to the parameters characterizing the transfer function in (C.7), the following remarks are in order. The constant k_W represents the input/output *steady-state gain*, which is equal to $1/H_0$ if $C(s)G(s)H_0$ has at least a pole at the origin. The *natural frequency* ω_n is the modulus of the complex conjugate poles, whose real part is given by $-\zeta\omega_n$ where ζ is the *damping ratio* of the pair of poles.

The influence of parameters ζ and ω_n on the closed-loop frequency response can be evaluated in terms of the resonance peak magnitude

$$M_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}},$$

occurring at the resonant frequency

$$\omega_r = \omega_n\sqrt{1-2\zeta^2},$$

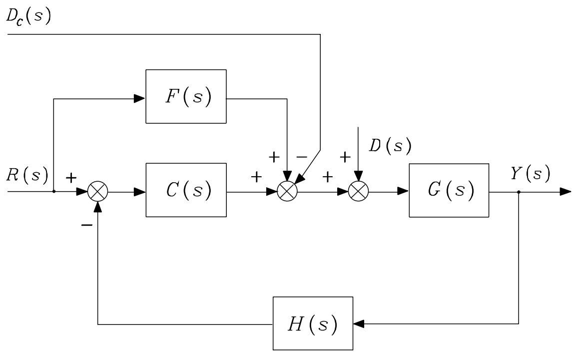

Fig. C.2. Feedback control structure with feedforward compensation

and of the 3 dB bandwidth

$$\omega_3 = \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{2 - 4\zeta^2 + 4\zeta^4}}.$$

A step input is typically used to characterize the transient response in the time domain. The influence of parameters ζ and ω_n on the *step response* can be evaluated in terms of the percentage of *overshoot*

$$s\% = 100 \exp(-\pi\zeta/\sqrt{1-\zeta^2}),$$

of the *rise time*

$$t_r \approx \frac{1.8}{\omega_n}$$

and of the *settling time* within 1%

$$t_s = \frac{4.6}{\zeta\omega_n}.$$

The adoption of a *feedforward compensation* action represents a feasible solution both for tracking a time-varying reference input and for enhancing rejection of the effects of a disturbance on the output. Consider the general scheme in Fig. C.2. Let $R(s)$ denote a given input reference and $D_c(s)$ denote a computed estimate of the disturbance $D(s)$; the introduction of the feedforward action yields the input/output relationship

$$Y(s) = \left(\frac{C(s)G(s)}{1 + C(s)G(s)H(s)} + \frac{F(s)G(s)}{1 + C(s)G(s)H(s)} \right) R(s) \qquad \text{(C.8)}$$
$$+ \frac{G(s)}{1 + C(s)G(s)H(s)} \big(D(s) - D_c(s) \big).$$

By assuming that the desired output is related to the reference input by a constant factor K_d and regarding the transducer as an instantaneous system ($H(s) \approx H_0 = 1/K_d$) for the current operating conditions, the choice

$$F(s) = \frac{K_d}{G(s)} \qquad \text{(C.9)}$$

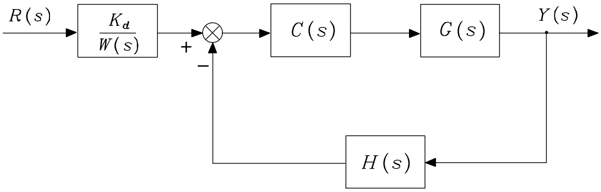

Fig. C.3. Feedback control structure with inverse model technique

yields the input/output relationship

$$Y(s) = Y_d(s) + \frac{G(s)}{1 + C(s)G(s)H_0}\big(D(s) - D_c(s)\big). \tag{C.10}$$

If $|C(j\omega)G(j\omega)H_0| \gg 1$, the effect of the disturbance on the output is further reduced by means of an accurate estimate of the disturbance.

Feedforward compensation technique may lead to a solution, termed *inverse model control*, illustrated in the scheme of Fig. C.3. It should be remarked, however, that such a solution is based on dynamics cancellation, and thus it can be employed only for a minimum-phase system, i.e., a system whose poles and zeros have all strictly negative real parts. Further, one should consider physical realizability issues as well as effects of parameter variations which prevent perfect cancellation.

C.2 Control of Nonlinear Mechanical Systems

If the system to control does not satisfy the linearity property, the control design problem becomes more complex. The fact that a *system* is qualified as *nonlinear*, whenever linearity does not hold, leads to understanding how it is not possible to resort to general techniques for control design, but it is necessary to face the problem for each class of nonlinear systems which can be defined through imposition of special properties.

On the above premise, the control design problem of nonlinear systems described by the dynamic model

$$\boldsymbol{H}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{u} \tag{C.11}$$

is considered, where $[\,\boldsymbol{x}^T \quad \dot{\boldsymbol{x}}^T\,]^T$ denotes the $(2n \times 1)$ *state* vector of the system, $\boldsymbol{u}$ is the $(n \times 1)$ *input* vector, $\boldsymbol{H}(\boldsymbol{x})$ is an $(n \times n)$ *positive definite* (and thus invertible) matrix depending on $\boldsymbol{x}$, and $\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ is an $(n \times 1)$ vector depending on state. Several *mechanical systems* can be reduced to this class, including manipulators with rigid links and joints.

The *control* law can be found through a nonlinear compensating action obtained by choosing the following *nonlinear state feedback* law (*inverse dynamics* control):

$$\boldsymbol{u} = \widehat{\boldsymbol{H}}(\boldsymbol{x})\boldsymbol{v} + \widehat{\boldsymbol{h}}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{C.12}$$

where $\widehat{\boldsymbol{H}}(\boldsymbol{x})$ and $\widehat{\boldsymbol{h}}(\boldsymbol{x})$ respectively denote the *estimates* of the terms $\boldsymbol{H}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$, computed on the basis of measures on the system state, and $\boldsymbol{v}$ is a new control input to be defined later. In general, it is

$$\widehat{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) + \Delta\boldsymbol{H}(\boldsymbol{x}) \tag{C.13}$$

$$\widehat{\boldsymbol{h}}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \Delta\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{C.14}$$

because of the unavoidable modelling approximations or as a consequence of an intentional simplification in the compensating action. Substituting (C.12) into (C.11) and accounting for (C.13), (C.14) yields

$$\ddot{\boldsymbol{x}} = \boldsymbol{v} + \boldsymbol{z}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{v}) \tag{C.15}$$

where

$$\boldsymbol{z}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{v}) = \boldsymbol{H}^{-1}(\boldsymbol{x})\big(\Delta\boldsymbol{H}(\boldsymbol{x})\boldsymbol{v} + \Delta\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}})\big).$$

If *tracking* of a trajectory $(\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t))$ is desired, the tracking error can be defined as

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{x}_d - \boldsymbol{x} \\ \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}} \end{bmatrix} \tag{C.16}$$

and it is necessary to derive the error dynamics equation to study convergence of the actual state to the desired one. To this end, the choice

$$\boldsymbol{v} = \ddot{\boldsymbol{x}}_d + \boldsymbol{w}(\boldsymbol{e}), \tag{C.17}$$

substituted into (C.15), leads to the error equation

$$\dot{\boldsymbol{e}} = \boldsymbol{F}\boldsymbol{e} - \boldsymbol{G}\boldsymbol{w}(\boldsymbol{e}) - \boldsymbol{G}\boldsymbol{z}(\boldsymbol{e}, \boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d), \tag{C.18}$$

where the $(2n \times 2n)$ and $(2n \times n)$ matrices, respectively,

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I} \end{bmatrix}$$

follow from the error definition in (C.16). Control law design consists of finding the error function $\boldsymbol{w}(\boldsymbol{e})$ which makes (C.18) *globally asymptotically stable*,[1] i.e.,

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

In the case of *perfect* nonlinear compensation ($\boldsymbol{z}(\cdot) = \boldsymbol{0}$), the simplest choice of the control action is the *linear* one

$$\begin{aligned} \boldsymbol{w}(\boldsymbol{e}) &= -\boldsymbol{K}_P(\boldsymbol{x}_d - \boldsymbol{x}) - \boldsymbol{K}_D(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) \\ &= [\,-\boldsymbol{K}_P \quad -\boldsymbol{K}_D\,]\,\boldsymbol{e}, \end{aligned} \tag{C.19}$$

[1] *Global* asymptotic stability is invoked to remark that the equilibrium state is asymptotically stable for any perturbation.

where asymptotic stability of the error equation is ensured by choosing *positive definite* matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$. The error transient behaviour is determined by the eigenvalues of the matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix} \tag{C.20}$$

characterizing the error dynamics

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e}. \tag{C.21}$$

If compensation is *imperfect*, then $\boldsymbol{z}(\cdot)$ cannot be neglected and the error equation in (C.18) takes on the general form

$$\dot{\boldsymbol{e}} = \boldsymbol{f}(\boldsymbol{e}). \tag{C.22}$$

It may be worth choosing the control law $\boldsymbol{w}(\boldsymbol{e})$ as the sum of a nonlinear term and a linear term of the kind in (C.19); in this case, the error equation can be written as

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{k}(\boldsymbol{e}), \tag{C.23}$$

where $\boldsymbol{A}$ is given by (C.20) and $\boldsymbol{k}(\boldsymbol{e})$ is available to make the system globally asymptotically stable. The equations in (C.22), (C.23) express nonlinear differential equations of the error. To test for stability and obtain advise on the choice of suitable control actions, one may resort to *Lyapunov direct method* illustrated below.

C.3 Lyapunov Direct Method

The philosophy of the *Lyapunov direct method* is the same as that of most methods used in control engineering to study stability, namely, testing for stability without solving the differential equations describing the dynamic system.

This method can be presented in short on the basis of the following reasoning. If it is possible to associate an energy-based description with a (linear or nonlinear) autonomous dynamic system and, for each system state with the exception of the equilibrium state, the time rate of such energy is negative, then energy decreases along any system trajectory until it attains its minimum at the equilibrium state; this argument justifies an intuitive concept of stability.

With reference to (C.22), by setting $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, the *equilibrium state* is $\boldsymbol{e} = \boldsymbol{0}$. A scalar function $V(\boldsymbol{e})$ of the system state, continuous together with its first derivative, is defined a *Lyapunov function* if the following properties hold:

$$V(\boldsymbol{e}) > 0 \qquad \forall \boldsymbol{e} \neq \boldsymbol{0}$$

$$\begin{aligned} V(\boldsymbol{e}) &= 0 & \boldsymbol{e} &= \boldsymbol{0} \\ \dot{V}(\boldsymbol{e}) &< 0 & \forall \boldsymbol{e} &\neq \boldsymbol{0} \\ V(\boldsymbol{e}) &\to \infty & \|\boldsymbol{e}\| &\to \infty. \end{aligned}$$

The existence of such a function ensures *global asymptotic stability* of the equilibrium $\boldsymbol{e} = \boldsymbol{0}$. In practice, the equilibrium $\boldsymbol{e} = \boldsymbol{0}$ is globally asymptotically stable if a positive definite, radially unbounded function $\boldsymbol{V}(\boldsymbol{e})$ is found so that its time derivative along the system trajectories is negative definite.

If positive definiteness of $V(\boldsymbol{e})$ is realized by the adoption of a *quadratic form*, i.e.,

$$V(\boldsymbol{e}) = \boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} \tag{C.24}$$

with $\boldsymbol{Q}$ a symmetric positive definite matrix, then in view of (C.22) it follows

$$\dot{V}(\boldsymbol{e}) = 2\boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{f}(\boldsymbol{e}). \tag{C.25}$$

If $\boldsymbol{f}(\boldsymbol{e})$ is so as to render the function $\dot{V}(\boldsymbol{e})$ negative definite, the function $V(\boldsymbol{e})$ is a *Lyapunov function*, since the choice (C.24) allows system global asymptotic stability to be proved. If $\dot{V}(\boldsymbol{e})$ in (C.25) is not negative definite for the given $V(\boldsymbol{e})$, nothing can be inferred on the stability of the system, since the Lyapunov method gives only a *sufficient* condition. In such cases one should resort to different choices of $V(\boldsymbol{e})$ in order to find, if possible, a negative definite $\dot{V}(\boldsymbol{e})$.

In the case when the property of negative definiteness does not hold, but $\dot{V}(\boldsymbol{e})$ is only *negative semi-definite*

$$\dot{V}(\boldsymbol{e}) \leq 0,$$

global asymptotic stability of the equilibrium state is ensured if the only system trajectory for which $\dot{V}(\boldsymbol{e})$ is *identically* null ($\dot{V}(\boldsymbol{e}) \equiv 0$) is the equilibrium trajectory $\boldsymbol{e} \equiv \boldsymbol{0}$ (a consequence of *La Salle theorem*).

Finally, consider the stability problem of the nonlinear system in the form (C.23); under the assumption that $\boldsymbol{k}(\boldsymbol{0}) = \boldsymbol{0}$, it is easy to verify that $\boldsymbol{e} = \boldsymbol{0}$ is an equilibrium state for the system. The choice of a Lyapunov function candidate as in (C.24) leads to the following expression for its derivative:

$$\dot{V}(\boldsymbol{e}) = \boldsymbol{e}^T (\boldsymbol{A}^T \boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A})\boldsymbol{e} + 2\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{k}(\boldsymbol{e}). \tag{C.26}$$

By setting

$$\boldsymbol{A}^T \boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A} = -\boldsymbol{P}, \tag{C.27}$$

the expression in (C.26) becomes

$$\dot{V}(\boldsymbol{e}) = -\boldsymbol{e}^T \boldsymbol{P}\boldsymbol{e} + 2\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{k}(\boldsymbol{e}). \tag{C.28}$$

The matrix equation in (C.27) is said to be a *Lyapunov equation*; for any choice of a symmetric positive definite matrix $\boldsymbol{P}$, the solution matrix $\boldsymbol{Q}$ exists and is symmetric positive definite if and only if the eigenvalues of $\boldsymbol{A}$ have all negative real parts. Since matrix $\boldsymbol{A}$ in (C.20) verifies such condition, it is always possible to assign a positive definite matrix $\boldsymbol{P}$ and find a positive definite matrix solution $\boldsymbol{Q}$ to (C.27). It follows that the first term on the right-hand side of (C.28) is negative definite and the stability problem is reduced to searching a control law so that $\boldsymbol{k}(\boldsymbol{e})$ renders the total $\dot{V}(\boldsymbol{e})$ negative (semi-)definite.

It should be underlined that La Salle theorem does not hold for *time-varying* systems (also termed *non-autonomous*) in the form

$$\dot{\boldsymbol{e}} = f(\boldsymbol{e}, t).$$

In this case, a conceptually analogous result which might be useful is the following, typically referred to as *Barbalat lemma* — of which it is indeed a consequence. Given a scalar function $V(\boldsymbol{e}, t)$ so that

1. $V(\boldsymbol{e}, t)$ is lower bounded
2. $\dot{V}(\boldsymbol{e}, t) \leq 0$
3. $\dot{V}(\boldsymbol{e}, t)$ is *uniformly continuous*

then it is $\lim_{t\to\infty} \dot{V}(\boldsymbol{e}, t) = 0$. Conditions 1 and 2 imply that $V(\boldsymbol{e}, t)$ has a bounded limit for $t \to \infty$. Since it is not easy to verify the property of uniform continuity from the definition, Condition 3 is usually replaced by

3'. $\ddot{V}(\boldsymbol{e}, t)$ is bounded

which is sufficient to guarantee validity of Condition 3. Barbalat lemma can obviously be used for time-invariant (autonomous) dynamic systems as an alternative to La Salle theorem, with respect to which some conditions are relaxed; in particular, $V(\boldsymbol{e})$ needs not necessarily be positive definite.

Bibliography

Linear systems analysis can be found in classical texts such as [61]. For the control of these systems see [82, 171]. For the analysis of nonlinear systems see [109]. Control of nonlinear mechanical systems is dealt with in [215].

D

Differential Geometry

The analysis of mechanical systems subject to nonholonomic constraints, such as wheeled mobile robots, requires some basic concepts of differential geometry and nonlinear controllability theory, that are briefly recalled in this appendix.

D.1 Vector Fields and Lie Brackets

For simplicity, the case of vectors $\boldsymbol{x} \in \mathbb{R}^n$ is considered. The tangent space at $\boldsymbol{x}$ (intuitively, the space of velocities of trajectories passing through $\boldsymbol{x}$) is hence denoted by $T_{\boldsymbol{x}}(\mathbb{R}^n)$. The presented notions are however valid in the more general case in which a *differentiable manifold* (i.e., a space that is locally diffeomorphic to $\mathbb{R}^n$) is considered in place of a Euclidean space.

A *vector field* $\boldsymbol{g} : \mathbb{R}^n \mapsto T_x(\mathbb{R}^n)$ is a mapping that assigns to each point $\boldsymbol{x} \in \mathbb{R}^n$ a tangent vector $\boldsymbol{g}(\boldsymbol{x}) \in T_x(\mathbb{R}^n)$. In the following it is always assumed that vector fields are *smooth*, i.e., such that the associated mappings are of class C^∞.

If the vector field $\boldsymbol{g}(\boldsymbol{x})$ is used to define a differential equation as in

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}), \tag{D.1}$$

the *flow* $\boldsymbol{\phi}_t^{\boldsymbol{g}}(\boldsymbol{x})$ of $\boldsymbol{g}$ is the mapping that associates to each point $\boldsymbol{x}$ the value at time t of the solution of (D.1) evolving from $\boldsymbol{x}$ at time 0, or

$$\frac{d}{dt}\boldsymbol{\phi}_t^{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g}(\phi_t^{\boldsymbol{g}}(\boldsymbol{x})). \tag{D.2}$$

The family of mappings $\{\boldsymbol{\phi}_t^{\boldsymbol{g}}\}$ is a one-parameter (i.e., t) group under the composition operator

$$\boldsymbol{\phi}_{t_1}^{\boldsymbol{g}} \circ \boldsymbol{\phi}_{t_2}^{\boldsymbol{g}} = \boldsymbol{\phi}_{t_1+t_2}^{\boldsymbol{g}}.$$

For example, for time-invariant linear systems it is $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and the flow is the linear operator $\boldsymbol{\phi}_t^{\boldsymbol{g}} = e^{\boldsymbol{A}t}$.

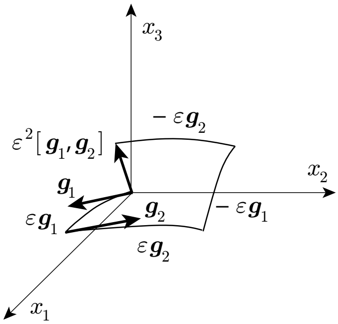

Fig. D.1. The net displacement of system (D.4) under the input sequence (D.5) is directed as the Lie bracket of the two vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$

Given two vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, the composition of their flows is non-commutative in general:

$$\phi_t^{\boldsymbol{g}_1} \circ \phi_s^{\boldsymbol{g}_2} \neq \phi_s^{\boldsymbol{g}_2} \circ \phi_t^{\boldsymbol{g}_1}.$$

The vector field $[\boldsymbol{g}_1, \boldsymbol{g}_2]$ defined as

$$[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x}) = \frac{\partial \boldsymbol{g}_2}{\partial x} \boldsymbol{g}_1(\boldsymbol{x}) - \frac{\partial \boldsymbol{g}_1}{\partial x} \boldsymbol{g}_2(\boldsymbol{x}) \tag{D.3}$$

is called *Lie bracket* of $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. The two vector field $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ *commute* if $[\boldsymbol{g}_1, \boldsymbol{g}_2] = 0$.

The Lie bracket operation has an interesting interpretation. Consider the driftless dynamic system

$$\dot{\boldsymbol{x}} = \boldsymbol{g}_1(\boldsymbol{x}) u_1 + \boldsymbol{g}_2(\boldsymbol{x}) u_2 \tag{D.4}$$

associated with the vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. If the inputs u_1 and u_2 are never active simultaneously, the solution of the differential equation (D.4) can be obtained by composing the flows of $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. In particular, consider the following input sequence:

$$u(t) = \begin{cases} u_1(t) = +1, u_2(t) = 0 & t \in [0, \varepsilon) \\ u_1(t) = 0, u_2(t) = +1 & t \in [\varepsilon, 2\varepsilon) \\ u_1(t) = -1, u_2(t) = 0 & t \in [2\varepsilon, 3\varepsilon) \\ u_1(t) = 0, u_2(t) = -1 & t \in [3\varepsilon, 4\varepsilon), \end{cases} \tag{D.5}$$

where ε is an infinitesimal time interval. The solution of (D.4) at time $t = 4\varepsilon$ can be obtained by following first the flow of $\boldsymbol{g}_1$, then of $\boldsymbol{g}_2$, then of $-\boldsymbol{g}_1$, and finally of $-\boldsymbol{g}_2$ (see Fig. D.1). By computing $\boldsymbol{x}(\varepsilon)$ through a series expansion at $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ along $\boldsymbol{g}_1$, then $\boldsymbol{x}(2\varepsilon)$ as a series expansion at $\boldsymbol{x}(\varepsilon)$ along $\boldsymbol{g}_2$, and so on, one obtains

$$\begin{aligned} \boldsymbol{x}(4\varepsilon) &= \phi_\varepsilon^{-\boldsymbol{g}_2} \circ \phi_\varepsilon^{-\boldsymbol{g}_1} \circ \phi_\varepsilon^{\boldsymbol{g}_2} \circ \phi_\varepsilon^{\boldsymbol{g}_1}(\boldsymbol{x}_0) \\ &= \boldsymbol{x}_0 + \varepsilon^2 \left(\frac{\partial \boldsymbol{g}_2}{\partial x} \boldsymbol{g}_1(\boldsymbol{x}_0) - \frac{\partial \boldsymbol{g}_1}{\partial x} \boldsymbol{g}_2(\boldsymbol{x}_0) \right) + O(\varepsilon^3). \end{aligned}$$

If $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ commute, the net displacement resulting from the input sequence (D.5) is zero.

The above expression shows that, at each point $\boldsymbol{x}$, infinitesimal motion of the driftless system (D.4) is possible not only in the directions belonging to the linear span of $\boldsymbol{g}_1(\boldsymbol{x})$ and $\boldsymbol{g}_2(\boldsymbol{x})$, but also in the direction of their Lie bracket $[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x})$. It can be proven that more complicated input sequences can be used to generate motion in the direction of higher-order Lie brackets, such as $[\boldsymbol{g}_1, [\boldsymbol{g}_1, \boldsymbol{g}_2]]$.

Similar constructive procedures can be given for systems with a *drift*[1] vector field, such as the following:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1(\boldsymbol{x})u_1 + \boldsymbol{g}_2(\boldsymbol{x})u_2. \tag{D.6}$$

Using appropriate input sequences, it is possible to generate motion in the direction of Lie brackets involving the vector field $\boldsymbol{f}$ as well as $\boldsymbol{g}_j$, $j = 1, 2$.

Example D.1

For a single-input linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{b}\,u,$$

the drift and input vector fields are $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{b}$, respectively. The following Lie brackets:

$$\begin{aligned} -[\boldsymbol{f}, \boldsymbol{g}] &= \boldsymbol{A}\boldsymbol{b} \\ [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}]] &= \boldsymbol{A}^2\boldsymbol{b} \\ -[\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}]]] &= \boldsymbol{A}^3\boldsymbol{b} \\ &\vdots \end{aligned}$$

represent well-known directions in which it is possible to move the system.

The *Lie derivative* of the scalar function $\alpha : \mathbb{R}^n \mapsto \mathbb{R}$ along vector field $\boldsymbol{g}$ is defined as

$$L_{\boldsymbol{g}}\,\alpha(\boldsymbol{x}) = \frac{\partial \alpha}{\partial \boldsymbol{x}}\,\boldsymbol{g}(\boldsymbol{x}). \tag{D.7}$$

The following properties of Lie brackets are useful in computation:

$$\begin{aligned} &[\boldsymbol{f}, \boldsymbol{g}] = -[\boldsymbol{g}, \boldsymbol{f}] && \text{(skew-symmetry)} \\ &[\boldsymbol{f}, [\boldsymbol{g}, \boldsymbol{h}]] + [\boldsymbol{h}, [\boldsymbol{f}, \boldsymbol{g}]] + [\boldsymbol{g}, [\boldsymbol{h}, \boldsymbol{f}]] = 0 && \text{(Jacobi identity)} \\ &[\alpha\boldsymbol{f}, \beta\boldsymbol{g}] = \alpha\beta[\boldsymbol{f}, \boldsymbol{g}] + \alpha(L_{\boldsymbol{f}}\beta)\boldsymbol{g} - \beta(L_{\boldsymbol{g}}\alpha)\boldsymbol{f} && \text{(chain rule)} \end{aligned}$$

[1] This term emphasizes how the presence of $\boldsymbol{f}$ will in general force the system to move ($\dot{\boldsymbol{x}} \neq \boldsymbol{0}$) even in the absence of inputs.

with $\alpha, \beta\colon \mathbb{R}^n \mapsto \mathbb{R}$. The vector space $\mathcal{V}(\mathbb{R}^n)$ of smooth vector fields on $\mathbb{R}^n$, equipped with the Lie bracket operation, is a *Lie algebra*.

The *distribution* Δ associated with the m vector fields $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$ is the mapping that assigns to each point $\boldsymbol{x} \in \mathbb{R}^n$ the subspace of $T_x(\mathbb{R}^n)$ defined as

$$\Delta(\boldsymbol{x}) = \text{span}\{\boldsymbol{g}_1(\boldsymbol{x}), \ldots, \boldsymbol{g}_m(\boldsymbol{x})\}. \tag{D.8}$$

Often, a shorthand notation is used:

$$\Delta = \text{span}\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}.$$

The distribution Δ is *nonsingular* if $\dim \Delta(\boldsymbol{x}) = r$, with r constant for all $\boldsymbol{x}$. In this case, r is called the *dimension* of the distribution. Moreover, Δ is called *involutive* if it is closed under the Lie bracket operation:

$$[\boldsymbol{g}_i, \boldsymbol{g}_j] \in \Delta \qquad \forall\, \boldsymbol{g}_i, \boldsymbol{g}_j \in \Delta.$$

The *involutive closure* $\bar{\Delta}$ of a distribution Δ is its closure under the Lie bracket operation. Hence, Δ is involutive if and only if $\bar{\Delta} = \Delta$. Note that the distribution $\Delta = \text{span}\{\boldsymbol{g}\}$ associated with a single vector field is always involutive, because $[\boldsymbol{g}, \boldsymbol{g}](\boldsymbol{x}) = \boldsymbol{0}$.

Example D.2

The distribution

$$\Delta = \text{span}\{\boldsymbol{g}_1, \boldsymbol{g}_2\} = \text{span}\left\{\begin{bmatrix}\cos x_3\\ \sin x_3\\ 0\end{bmatrix}, \begin{pmatrix}0\\ 0\\ 1\end{pmatrix}\right\}$$

is nonsingular and has dimension 2. It is not involutive, because the Lie bracket

$$[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x}) = \begin{bmatrix}\sin x_3\\ -\cos x_3\\ 0\end{bmatrix}$$

is always linearly independent of $\boldsymbol{g}_1(\boldsymbol{x})$ and $\boldsymbol{g}_2(\boldsymbol{x})$. Its involutive closure is therefore

$$\bar{\Delta} = \text{span}\{\boldsymbol{g}_1, \boldsymbol{g}_2, [\boldsymbol{g}_1, \boldsymbol{g}_2]\}.$$

D.2 Nonlinear Controllability

Consider a nonlinear dynamic system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{j=1}^{m} \boldsymbol{g}_j(\boldsymbol{x}) u_j, \tag{D.9}$$

that is called *affine* in the inputs u_j. The state $\boldsymbol{x}$ takes values in $\mathbb{R}^n$, while each component u_j of the control input $\boldsymbol{u} \in \mathbb{R}^m$ takes values in the class $\mathcal{U}$ of piecewise-constant functions.

Denote by $\boldsymbol{x}(t, 0, \boldsymbol{x}_0, \boldsymbol{u})$ the solution of (D.9) at time $t \geq 0$, corresponding to an input $\boldsymbol{u}\colon [0, t] \to \mathcal{U}$ and an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Such a solution exists and is unique provided that the drift vector field $\boldsymbol{f}$ and the input vector fields $\boldsymbol{g}_j$ are of class C^∞. System (D.9) is said to be *controllable* if, for any choice of $\boldsymbol{x}_1, \boldsymbol{x}_2$ in $\mathbb{R}^n$, there exists a time instant T and an input $\boldsymbol{u}\colon [0, T] \to \mathcal{U}$ such that $\boldsymbol{x}(T, 0, \boldsymbol{x}_1, \boldsymbol{u}) = \boldsymbol{x}_2$.

The *accessibility algebra* $\mathcal{A}$ of system (D.9) is the smallest subalgebra of $\mathcal{V}(\mathbb{R}^n)$ that contains $\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m$. By definition, all the Lie brackets that can be generated using these vector fields belong to $\mathcal{A}$. The *accessibility distribution* $\Delta_{\mathcal{A}}$ of system (D.9) is defined as

$$\Delta_{\mathcal{A}} = \mathrm{span}\{\boldsymbol{v} | \boldsymbol{v} \in \mathcal{A}\}. \tag{D.10}$$

In other words, $\Delta_{\mathcal{A}}$ is the involutive closure of $\Delta = \mathrm{span}\{\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$.

The computation of $\Delta_{\mathcal{A}}$ may be organized as an iterative procedure

$$\Delta_{\mathcal{A}} = \mathrm{span}\left\{\boldsymbol{v} | \boldsymbol{v} \in \Delta_i, \forall i \geq 1\right\},$$

with

$$\begin{aligned} \Delta_1 &= \Delta = \mathrm{span}\{\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\} \\ \Delta_i &= \Delta_{i-1} + \mathrm{span}\{[\boldsymbol{g}, \boldsymbol{v}] |\, \boldsymbol{g} \in \Delta_1, \boldsymbol{v} \in \Delta_{i-1}\}, \quad i \geq 2. \end{aligned}$$

This procedure stops after κ steps, where κ is the smallest integer such that $\Delta_{\kappa+1} = \Delta_\kappa = \Delta_{\mathcal{A}}$. This number is called the *nonholonomy degree* of the system and is related to the 'level' of Lie brackets that must be included in $\Delta_{\mathcal{A}}$. Since $\dim \Delta_{\mathcal{A}} \leq n$, it is $\kappa \leq n - m$ necessarily.

If system (D.9) is driftless

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x}) u_i, \tag{D.11}$$

the accessibility distribution $\Delta_{\mathcal{A}}$ associated with vector fields $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m$ characterizes its controllability. In particular, system (D.11) is controllable if and only if the following *accessibility rank condition* holds:

$$\dim \Delta_{\mathcal{A}}(\boldsymbol{x}) = n. \tag{D.12}$$

Note that for driftless systems the iterative procedure for building $\Delta_{\mathcal{A}}$ starts with $\Delta_1 = \Delta = \mathrm{span}\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$, and therefore $\kappa \leq n - m + 1$.

For systems in the general form (D.9), condition (D.12) is only necessary for controllability. There are, however, two notable exceptions:

- If system (D.11) is controllable, the system with drift obtained by performing a *dynamic extension* of (D.11)

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x}) v_i \tag{D.13}$$

$$\dot{v}_i = u_i, \qquad i = 1, \ldots, m, \tag{D.14}$$

  i.e., by adding an integrator on each input channel, is also controllable.
- For a linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \sum_{j=1}^{m} \boldsymbol{b}_j u_j = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

  (D.12) becomes

$$\varrho\left(\begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2\boldsymbol{B} & \ldots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix}\right) = n, \tag{D.15}$$

  i.e., the well-known necessary and sufficient condition for controllability due to Kalman.

Bibliography

The concepts briefly recalled in this appendix can be studied in detail in various tests of differential geometry [94, 20] and nonlinear control theory [104, 168, 195].

E

Graph Search Algorithms

This appendix summarizes some basic concepts on algorithm complexity and graph search techniques that are useful in the study of motion planning.

E.1 Complexity

A major criterion for assessing the efficiency of an algorithm A is its *running time*, i.e., the time needed for executing the algorithm in a computational model capturing the most relevant characteristics of an actual elaboration system. In practice, one is interested in estimating the running time as a function of a single parameter n characterizing the *size* of the input within a specific class of instances of the problem. In motion planning, this parameter may be the dimension of the configuration space, or the number of vertices of the free configuration space (if it is a polygonal subset).

In *worst-case analysis*, $t(n)$ denotes the maximum running time of A in correspondence of input instances of size n. Other kinds of analyses (e.g., average-case) are possible but they are less critical or general, requiring a statistical knowledge of the input distribution that may not be available.

The exact functional expression of $t(n)$ depends on the implementation of the algorithm, and is of little practical interest because the running time in the adopted computational model is only an approximation of the actual one. More significant is the *asymptotic behaviour* of $t(n)$, i.e., the rate of growth of $t(n)$ with n. Denote by $O(f(n))$ the set of real functions $g(n)$ such that

$$c_1 f(n) \leq g(n) \leq c_2 f(n) \qquad \forall\, n \geq n_0,$$

with c_1, c_2 and n_0 positive constants. If the worst-case running time of A is $O(f(n))$, i.e., if $t(n) \in O(f(n))$, the *time complexity* of A is said to be $O(f(n))$.

A very important class is represented by algorithms whose worst-case running time is asymptotically polynomial in the size of the input. In particular, if $t(n) \in O(n^p)$, for some $p \geq 0$, the algorithm is said to have *polynomial* time complexity. If the asymptotic behaviour of the worst-case running time is not polynomial, the time complexity of the algorithm is *exponential*. Note that here 'exponential' actually means 'not bounded by any polynomial function'.

The asymptotic behaviour of an algorithm with exponential time complexity is such that in the worst case it can only be applied to problems of 'small' size. However, there exist algorithms of exponential complexity that are very efficient on average, i.e., for the most frequent classes of input. A well known example is the simplex algorithm for solving linear programming problems. Similarly, there are algorithms with polynomial time complexity which are inefficient in practice because c_1, c_2 or p are 'large'.

The above concepts can be extended to inputs whose size is characterized by more than one parameter, or to performance criteria different from running time. For example, the memory space required by an algorithm is another important measure. The *space complexity* of an algorithm is said to be $O(f(n))$ if the memory space required for its execution is a function in $O(f(n))$.

E.2 Breadth-first and Depth-first Search

Let $G = (N, A)$ be a graph consisting of a set N of nodes and a set A of arcs, with cardinality n and a respectively. It is assumed that G is represented by an *adjacency list*: to each node N_i is associated a list of nodes that are connected to N_i by an arc. Consider the problem of searching G to find a path from a start node N_s to a goal node N_g. The simplest graph search strategies are *breadth-first search* (BFS) and *depth-first search* (DFS). These are briefly described in the following with reference to an iterative implementation.

Breadth-first search makes use of a *queue* — i.e., a FIFO (First In First Out) data structure — of nodes called OPEN. Initially, OPEN contains only the start node N_s, which is marked *visited*. All the other nodes in G are marked *unvisited*. At each iteration the first node in OPEN is extracted, and all its *unvisited* adjacent nodes are marked *visited* and inserted in OPEN. The search terminates when either N_g is inserted in OPEN or OPEN is empty (failure). During the search, the algorithm maintains the *BFS tree*, which contains only those arcs that have led to discovering *unvisited* nodes. This tree contains one and only one path connecting the start node to each visited node, and hence also a solution path from N_s to N_g, if it exists.

In depth-first search, OPEN is a *stack*, i.e., a LIFO (Last In First Out) data structure. Like in the breadth-first case, it contains initially only the start node N_s marked *visited*. When a node N_j is inserted in OPEN, the node N_i which has determined its insertion is memorized. At each iteration, the first node in OPEN is extracted. If it is *unvisited*, it is marked *visited* and the arc connecting N_i to N_j is inserted in the *DFS tree*. All *unvisited* nodes that are adjacent to N_j are inserted in OPEN. The search terminates when either N_g is inserted in OPEN or OPEN is empty (failure). Like in the BFS, the DFS tree contains the solution path from N_s to N_g, if it exists.

Both breadth-first and depth-first search have time complexity $O(a)$. Note that BFS and DFS are actually *traversal* strategies, because they do not use any information about the goal node; the graph is simply traversed until N_g is marked *visited*. Both the algorithms are *complete*, i.e., they find a solution path if it exists and report failure otherwise.

E.3 $A^\star$ Algorithm

In many applications, the arcs of G are labelled with positive numbers called *weights*. As a consequence, one may define the *cost* of a path on G as the sum of the weights of its arcs. Consider the problem of connecting N_s to N_g on G through a path of minimum cost, simply called *minimum path*. In motion planning problems, for example, the nodes generally represent points in configuration space, and it is then natural to define the weight of an arc as the length of the path that it represents. The minimum path is obviously interesting because it is the shortest among those joining N_s to N_g on G.

A widely used strategy for determining the minimum path on a graph is the $A^\star$ algorithm. $A^\star$ visits the nodes of G iteratively starting from N_s, storing only the current minimum paths from N_s to the visited nodes in a tree T. The algorithm employs a cost function $f(N_i)$ for each node N_i visited during the search. This function, which is an *estimate* of the cost of the minimum path that connects N_s to N_g passing through N_i, is computed as

$$f(N_i) = g(N_i) + h(N_i),$$

where $g(N_i)$ is the cost of the path from N_s to N_i as stored in the current tree T, and $h(N_i)$ is a *heuristic* estimate of the cost $h^\star(N_i)$ of the minimum path between N_i and N_g. While the value of $g(N_i)$ is uniquely determined by the search, any choice of $h(\cdot)$ such that

$$\forall N_i \in N : 0 \leq h(N_i) \leq h^\star(N_i) \tag{E.1}$$

is admissible. Condition (E.1) means that $h(\cdot)$ must not 'overestimate' the cost of the minimum path from N_i to N_g.

In the following, a pseudocode description of $A^\star$ is given. For its understanding, some preliminary remarks are needed:

- all the nodes are initially *unvisited*, except N_s which is *visited*;
- at the beginning, T contains only N_s;
- OPEN is a list of nodes that initially contains only N_s;
- N_{best} is the node in OPEN with the minimum value of f (in particular, it is the first node if OPEN is sorted by increasing values of f);
- $\text{ADJ}(N_i)$ is the adjacency list of N_i;
- $c(N_i, N_j)$ is the weight of the arc connecting N_i to N_j.

$A^\star$ algorithm

repeat
find and extract N_{best} from OPEN
if $N_{\text{best}} = N_g$ **then** exit
for each node N_i in $\text{ADJ}(N_{\text{best}})$ **do**
if N_i is *unvisited* **then**
add N_i to T with a pointer toward N_{best}
insert N_i in OPEN; mark N_i *visited*
else if $g(N_{\text{best}}) + c(N_{\text{best}}, N_i) < g(N_i)$ **then**
redirect the pointer of N_i in T toward N_{best}
if N_i is not in OPEN **then**
insert N_i in OPEN
else update $f(N_i)$
end if
end if
until OPEN is empty

Under condition (E.1), the $A^\star$ algorithm is complete. In particular, if the algorithm terminates with an empty OPEN, there exists no path in G from N_s to N_g (failure); otherwise, the tree T contains the minimum path from N_s to N_g, which can be reconstructed by backtracking from N_g to N_s.

The $A^\star$ algorithm with the particular (admissible) choice $h(N_i) = 0$, for each node N_i, is equivalent to the *Dijkstra algorithm*. If the nodes in G represent points in a Euclidean space, an admissible heuristic is the Euclidean distance between N and N_g. In fact, the length of the minimum path between N_i and N_g is bounded below by the Euclidean distance.

The extraction of a node from OPEN and the visit of its adjacent nodes is called node *expansion*. Given two admissible heuristic functions h_1 and h_2 such that $h_2(N_i) \geq h_1(N_i)$, for each node N_i in G, it is possible to prove that each node in G expanded by $A^\star$ using h_2 is also expanded using h_1. This means that $A^\star$ equipped with the heuristic h_2 is *at least* as efficient as $A^\star$ equipped with the heuristic h_1; h_2 is said to be *more informed* than h_1.

The $A^\star$ algorithm can be implemented with time complexity $O(a \log n)$.

Bibliography

The notions briefly recalled in this appendix are explained in detail in various texts on algorithm theory and artificial intelligence, such as [51, 189, 202].

References

1. C. Abdallah, D. Dawson, P. Dorato, M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24–30, 1991.
2. M. Aicardi, G. Casalino, A. Bicchi, A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.
3. J.S. Albus, H.G. McCain, R. Lumia, *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, NBS tech. note 1235, Gaithersburg, MD, 1987.
4. C.H. An, C.G. Atkeson, J.M. Hollerbach, *Model-Based Control of a Robot Manipulator*, MIT Press, Cambridge, MA, 1988.
5. R.J. Anderson, M.W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 549–556, 1988.
6. J. Angeles, *Spatial Kinematic Chains: Analysis, Synthesis, Optimization*, Springer-Verlag, Berlin, 1982.
7. S. Arimoto, F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," in *Robotics Research: The First International Symposium*, M. Brady, R. Paul (Eds.), MIT Press, Cambridge, MA, pp. 783–799, 1984.
8. R.C. Arkin, *Behavior-Based Robotics*, MIT Press, Cambridge, MA, 1998.
9. B. Armstrong-Hélouvry, *Control of Machines with Friction*, Kluwer, Boston, MA, 1991.
10. H. Asada, J.-J.E. Slotine, *Robot Analysis and Control*, Wiley, New York, 1986.
11. H. Asada, K. Youcef-Toumi, "Analysis and design of a direct-drive arm with a five-bar-link parallel drive mechanism," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 225–230, 1984.
12. H. Asada, K. Youcef-Toumi, *Direct-Drive Robots*, MIT Press, Cambridge, MA, 1987.
13. C.G. Atkeson, C.H. An, J.M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.
14. J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proc. 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, pp. 722–728, 1985.

15. A. Balestrino, G. De Maria, L. Sciavicco, “An adaptive model following control for robotic manipulators,” *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 105, pp. 143–151, 1983.
16. A. Balestrino, G. De Maria, L. Sciavicco, B. Siciliano, “An algorithmic approach to coordinate transformation for robotic manipulators,” *Advanced Robotics*, vol. 2, pp. 327–344, 1988.
17. J. Barraquand, J.-C. Latombe, “Robot motion planning: A distributed representation approach,” *International Journal of Robotics Research*, vol. 10, pp. 628–649, 1991.
18. G. Bastin, G. Campion, B. D’Andréa-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 47–62, 1996.
19. A.K. Bejczy, *Robot Arm Dynamics and Control*, memo. TM 33-669, Jet Propulsion Laboratory, California Institute of Technology, 1974.
20. W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, Orlando, FL, 1986.
21. J. Borenstein, H.R. Everett, L. Feng, *Navigating Mobile Robots: Systems and Techniques*, A K Peters, Wellesley, MA, 1996.
22. B.K.K. Bose, *Modern Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, NJ, 2001.
23. O. Bottema, B. Roth, *Theoretical Kinematics*, North Holland, Amsterdam, 1979.
24. T.L. Boullion, P.L. Odell, *Generalized Inverse Matrices*, Wiley, New York, 1971.
25. M. Brady, “Artificial intelligence and robotics,” *Artificial Intelligence*, vol. 26, pp. 79–121, 1985.
26. M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Pérez, M.T. Mason, (Eds.), *Robot Motion: Planning and Control*, MIT Press, Cambridge, MA, 1982.
27. H. Bruyninckx, J. De Schutter, “Specification of force-controlled actions in the “task frame formalism” — A synthesis,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 581–589, 1996.
28. H. Bruyninckx, S. Dumey, S. Dutré, J. De Schutter, “Kinematic models for model-based compliant motion in the presence of uncertainty,” *International Journal of Robotics Research*, vol. 14, pp. 465–482, 1995.
29. F. Caccavale, P. Chiacchio, “Identification of dynamic parameters and feedforward control for a conventional industrial manipulator,” *Control Engineering Practice*, vol. 2, pp. 1039–1050, 1994.
30. F. Caccavale, C. Natale, B. Siciliano, L. Villani, “Resolved-acceleration control of robot manipulators: A critical review with experiments,” *Robotica*, vol. 16, pp. 565–573, 1998.
31. F. Caccavale, C. Natale, B. Siciliano, L. Villani, “Six-DOF impedance control based on angle/axis representations,” *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 289–300, 1999.
32. F. Caccavale, C. Natale, B. Siciliano, L. Villani, ”Robot impedance control with nondiagonal stiffness,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 1943–1946, 1999.
33. J.F. Canny, *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA, 1988.

34. C. Canudas de Wit, H. Khennouf, C. Samson, O.J. Sørdalen, “Nonlinear control design for mobile robots,” in *Recent Trends in Mobile Robots*, Y.F. Zheng, (Ed.), pp. 121–156, World Scientific Publisher, Singapore, 1993.
35. F. Chaumette, “Image moments: A general and useful set of features for visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 1116-1127, 2005.
36. F. Chaumette, S. Hutchinson, “Visual servo control. Part I: Basic approaches,” *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
37. P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, “Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy,” *International Journal of Robotics Research*, vol. 10, pp. 410–425, 1991.
38. P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, “Influence of gravity on the manipulability ellipsoid for robot arms,” *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 723–727, 1992.
39. P. Chiacchio, F. Pierrot, L. Sciavicco, B. Siciliano, “Robust design of independent joint controllers with experimentation on a high-speed parallel robot,” *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 393–403, 1993.
40. S. Chiaverini, L. Sciavicco, “The parallel approach to force/position control of robotic manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 4, pp. 361–373, 1993.
41. S. Chiaverini, B. Siciliano, “The unit quaternion: A useful tool for inverse kinematics of robot manipulators,” *Systems Analysis Modelling Simulation*, vol. 35, pp. 45–60, 1999.
42. S. Chiaverini, B. Siciliano, O. Egeland, “Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator,” *IEEE Transactions on Control Systems Technology*, vol. 2, pp. 123–134, 1994.
43. S. Chiaverini, B. Siciliano, L. Villani, “Force/position regulation of compliant robot manipulators,” *IEEE Transactions on Automatic Control*, vol. 39, pp. 647–652, 1994.
44. S.L. Chiu, “Task compatibility of manipulator postures,” *International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.
45. H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.
46. J.C.K. Chou, “Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 53–64, 1992.
47. A.I. Comport, E. Marchand, M. Pressigout, F. Chaumette, “Real-time markerless tracking for augmented reality: The virtual visual servoing framework,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 615–628, 2006.
48. P.I. Corke, *Visual Control of Robots: High-Performance Visual Servoing*, Research Studies Press, Taunton, UK, 1996.
49. P. Corke, S. Hutchinson, “A new partitioned approach to image-based visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 507-515, 2001.
50. M. Corless, G. Leitmann, “Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems,” *IEEE Transactions on Automatic Control*, vol. 26, pp. 1139–1144, 1981.

51. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.
52. J.J. Craig, *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, Reading, MA, 1988.
53. J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed., Pearson Prentice Hall, Upper Saddle River, NJ, 2004.
54. C. De Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
55. T.L. De Fazio, D.S. Seltzer, D.E. Whitney, "The instrumented Remote Center of Compliance," *Industrial Robot*, vol. 11, pp. 238–242, 1984.
56. A. De Luca, *A Spline Generator for Robot Arms*, tech. rep. RAL 68, Rensselaer Polytechnic Institute, Department of Electrical, Computer, and Systems Engineering, 1986.
57. A. De Luca, C. Manes, "Modeling robots in contact with a dynamic environment," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 542–548, 1994.
58. A. De Luca, G. Oriolo, C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond, (Ed.), Springer-Verlag, Berlin, Germany, 1998.
59. A. De Luca, G. Oriolo, B. Siciliano, "Robot redundancy resolution at the acceleration level," *Laboratory Robotics and Automation*, vol. 4, pp. 97–106, 1992.
60. J. Denavit, R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
61. P.M. DeRusso, R.J. Roy, C.M. Close, A.A. Desrochers, *State Variables for Engineers*, 2nd ed., Wiley, New York, 1998.
62. J. De Schutter, H. Bruyninckx, S. Dutré, J. De Geeter, J. Katupitiya, S. Demey, T. Lefebvre, "Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motions," *International Journal of Robotics Research*, vol. 18, pp. 1161–1184, 1999.
63. J. De Schutter, H. Bruyninckx, W.-H. Zhu, M.W. Spong, "Force control: A bird's eye view," in *Control Problems in Robotics and Automation*, B. Siciliano, K.P. Valavanis, (Ed.), pp. 1–17, Springer-Verlag, London, UK, 1998.
64. J. De Schutter, H. Van Brussel, "Compliant robot motion I. A formalism for specifying compliant motion tasks," *International Journal of Robotics Research*, vol. 7, no. 4, pp. 3–17, 1988.
65. J. De Schutter, H. Van Brussel, "Compliant robot motion II. A control approach based on external control loops," *International Journal of Robotics Research*, vol. 7, no. 4, pp. 18–33, 1988.
66. K.L. Doty, C. Melchiorri, C. Bonivento, "A theory of generalized inverses applied to robotics," *International Journal of Robotics Research*, vol. 12, pp. 1–19, 1993.
67. S. Dubowsky, D.T. DesForges, "The application of model referenced adaptive control to robotic manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 101, pp. 193–200, 1979.
68. C. Edwards, L. Galloway, "A single-point calibration technique for a six-degree–of–freedom articulated arm," *International Journal of Robotics Research*, vol. 13, pp. 189–199, 1994.
69. O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 471–475, 1987.

70. S.D. Eppinger, W.P. Seering, "Introduction to dynamic models for robot force control," *IEEE Control Systems Magazine*, vol. 7, no. 2, pp. 48–52, 1987.
71. B. Espiau, F. Chaumette, P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 313–326, 1992.
72. H.R. Everett, *Sensors for Mobile Robots: Theory and Application*, AK Peters, Wellesley, MA, 1995.
73. G.E. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed., Morgan Kaufmann Publishers, San Francisco, CA, 2001.
74. E.D. Fasse, P.C. Breedveld, "Modelling of elastically coupled bodies: Parts I–II", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 120, pp. 496–506, 1998.
75. O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Boston, MA, 1993.
76. R. Featherstone, "Position and velocity transformations between robot end-effector coordinates and joint angles," *International Journal of Robotics Research*, vol. 2, no. 2, pp. 35–45, 1983.
77. R. Featherstone, *Robot Dynamics Algorithms*, Kluwer, Boston, MA, 1987.
78. R. Featherstone, O. Khatib, "Load independence of the dynamically consistent inverse of the Jacobian matrix," *International Journal of Robotics Research*, vol. 16, pp. 168–170, 1997.
79. J. Feddema, O. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 691–700, 1989.
80. M. Fliess, J. Lévine, P. Martin, P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.
81. J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, Springer, New York, 2004.
82. G.F. Franklin, J.D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed., Prentice-Hall, Lebanon, IN, 2005.
83. E. Freund, "Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 65–78, 1982.
84. L.-C. Fu, T.-L. Liao, "Globally stable robust tracking of nonlinear systems using variable structure control with an application to a robotic manipulator," *IEEE Transactions on Automatic Control*, vol. 35, pp. 1345–1350, 1990.
85. M. Gautier, W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 368–373, 1990.
86. A.A. Goldenberg, B. Benhabib, R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal of Robotics and Automation*, vol. 1, pp. 14–20, 1985.
87. H. Goldstein, C.P. Poole, J.L. Safko, *Classical Mechanics*, 3rd ed., Addison-Wesley, Reading, MA, 2002.
88. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
89. M.C. Good, L.M. Sweet, K.L. Strobel, "Dynamic models for control system design of integrated robot and drive systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 53–59, 1985.

90. D.M. Gorinevski, A.M. Formalsky, A.Yu. Schneider, *Force Control of Robotics Systems*, CRC Press, Boca Raton, FL, 1997.
91. W.A. Gruver, B.I. Soroka, J.J. Craig, T.L. Turner, "Industrial robot programming languages: A comparative evaluation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 14, pp. 565–570, 1984.
92. G. Hager, W. Chang, A. Morse, "Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 30–39, 1995.
93. R.M. Haralick, L.G. Shapiro, *Computer and Robot Vision*, vols. 1 & 2, Addison-Wesley, Reading, MA, 1993.
94. S. Helgason, *Differential Geometry and Symmetric Spaces*, Academic Press, New York, NY, 1962.
95. N. Hogan, "Impedance control: An approach to manipulation: Part I — Theory," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–7, 1985.
96. J.M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, pp. 730–736, 1980.
97. J.M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 102–106, 1984.
98. J.M. Hollerbach, "A survey of kinematic calibration," in *The Robotics Review 1*, O. Khatib, J.J. Craig, and T. Lozano-Pérez (Eds.), MIT Press, Cambridge, MA, pp. 207–242, 1989.
99. J.M. Hollerbach, G. Sahar, "Wrist-partitioned inverse kinematic accelerations and manipulator dynamics," *International Journal of Robotics Research*, vol. 2, no. 4, pp. 61–76, 1983.
100. R. Horowitz, M. Tomizuka, "An adaptive control scheme for mechanical manipulators — Compensation of nonlinearity and decoupling control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 127–135, 1986.
101. T.C.S. Hsia, T.A. Lasky, Z. Guo, "Robust independent joint controller design for industrial robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 38, pp. 21–25, 1991.
102. P. Hsu, J. Hauser, S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, pp. 133–148, 1989.
103. S. Hutchinson, G. Hager, P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651–670, 1996.
104. A. Isidori, *Nonlinear Control Systems*, 3rd ed., Springer-Verlag, London, UK, 1995.
105. H. Kazerooni, P.K. Houpt, T.B. Sheridan, "Robust compliant motion of manipulators, Part I: The fundamental concepts of compliant motion," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 83–92, 1986.
106. J.L. Jones, A.M. Flynn, *Mobile Robots: Inspiration to Implementation*, AK Peters, Wellesley, MA, 1993.
107. L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1996.

108. R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, pp. 39–48, 2000.
109. H.K. Khalil, *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
110. W. Khalil, F. Bennis, "Symbolic calculation of the base inertial parameters of closed-loop robots," *International Journal of Robotics Research*, vol. 14, pp. 112–128, 1995.
111. W. Khalil, E. Dombre, *Modeling, Identification and Control of Robots*, Hermes Penton Ltd, London, 2002.
112. W. Khalil, J.F. Kleinfinger, "Minimum operations and minimum parameters of the dynamic model of tree structure robots," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 517–526, 1987.
113. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
114. O. Khatib, "A unified approach to motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 43–53, 1987.
115. P.K. Khosla, "Categorization of parameters in the dynamic robot model," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 261–268, 1989.
116. P.K. Khosla, T. Kanade, "Parameter identification of robot dynamics," in *Proceedings of 24th IEEE Conference on Decision and Control*, Fort Lauderdale, FL, pp. 1754–1760, 1985.
117. P.K. Khosla, T. Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators," *International Journal of Robotics Research*, vol. 7, no. 1, pp. 18–28, 1988.
118. C.A. Klein, C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 245–250, 1983.
119. D.E. Koditschek, "Natural motion for robot arms," *Proc. 23th IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 733–735, 1984.
120. A.J. Koivo, *Fundamentals for Control of Robotic Manipulators*, Wiley, New York, 1989.
121. K. Kreutz, "On manipulator control by exact linearization," *IEEE Transactions on Automatic Control*, vol. 34, pp. 763–767, 1989.
122. J.-C. Latombe, *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
123. J.-P. Laumond, (Ed.), *Robot Motion Planning and Control*, Springer-Verlag, Berlin, 1998.
124. S.M. LaValle, *Planning Algorithms*, Cambridge University Press, New York, 2006.
125. S.M. LaValle, J.J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *New Directions in Algorithmic and Computational Robotics*, B.R. Donald, K. Lynch, D. Rus, (Eds.), AK Peters, Wellesley, MA, pp. 293–308, 2001.
126. M.B. Leahy, G.N. Saridis, "Compensation of industrial manipulator dynamics," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 73–84, 1989.
127. C.S.G. Lee, "Robot kinematics, dynamics and control," *IEEE Computer*, vol. 15, no. 12, pp. 62–80, 1982.
128. W. Leonhard, *Control of Electrical Drives*, Springer-Verlag, New York, 2001.

129. A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 868–871, 1977.
130. K.Y. Lim, M. Eslami, "Robust adaptive controller designs for robot manipulator systems," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 54–66, 1987.
131. C.S. Lin, P.R. Chang, J.Y.S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on Automatic Control*, vol. 28, pp. 1066–1073, 1983.
132. S.K. Lin, "Singularity of a nonlinear feedback control scheme for robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 134–139, 1989.
133. H. Lipkin, J. Duffy, "Hybrid twist and wrench control for a robotic manipulator," *ASME Journal of Mechanism, Transmissions, and Automation Design*, vol. 110, pp. 138–144, 1988.
134. V. Lippiello, B. Siciliano, L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Transactions on Robotics*, vol. 23, pp. 73–86, 2007.
135. D.A. Lizárraga, "Obstructions to the existence of universal stabilizers for smooth control systems," *Mathematics of Control, Signals, and Systems*, vol. 16, pp. 255–277, 2004.
136. J. Lončarić, "Normal forms of stiffness and compliance matrices," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 567–572, 1987.
137. T. Lozano-Pérez, "Automatic planning of manipulator transfer movements," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 681–698, 1981.
138. T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computing*, vol. 32, pp. 108–120, 1983.
139. T. Lozano-Pérez, "Robot programming," *Proceedings IEEE*, vol. 71, pp. 821–841, 1983.
140. T. Lozano-Pérez, M.T. Mason, R.H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
141. J.Y.S. Luh, "Conventional controller design for industrial robots: A tutorial," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 298–316, 1983.
142. J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "On-line computational scheme for mechanical manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, 1980.
143. J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, pp. 468–474, 1980.
144. J.Y.S. Luh, Y.-F. Zheng, "Computation of input generalized forces for robots with closed kinematic chain mechanisms," *IEEE Journal of Robotics and Automation* vol. 1, pp. 95–103, 1985.
145. V.J. Lumelsky, *Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World*, Wiley, Hoboken, NJ, 2006.
146. Y. Ma, S. Soatto, J. Kosecka, S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, New York, 2003.

147. A.A. Maciejewski, C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
148. E. Malis, F. Chaumette, S. Boudet, "2-1/2D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 238–250, 1999.
149. B.R. Markiewicz, *Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator*, memo. TM 33-601, JPL, Pasadena, CA, 1973.
150. M.T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 418–432, 1981.
151. J.M. McCarthy, *An Introduction to Theoretical Kinematics*, MIT Press, Cambridge, MA, 1990.
152. N.H. McClamroch, D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Transactions on Automatic Control*, vol. 33, pp. 419–426, 1988.
153. R.T. M'Closkey, R.M. Murray, "Exponential stabilization of driftless nonlinear control systems using homogeneous feedback," *IEEE Transactions on Automatic Control*, vol. 42, pp. 614–628, 1997.
154. L. Meirovitch, *Dynamics and Control of Structures*, Wiley, New York, 1990.
155. C. Melchiorri, *Traiettorie per Azionamenti Elettrici*, Progetto Leonardo, Bologna, I, 2000.
156. N. Manring, *Hydraulic Control Systems*, Wiley, New York, 2005.
157. R. Middleton, G.C. Goodwin, "Adaptive computed torque control for rigid link manipulators," *Systems & Control Letters*, vol. 10, pp. 9–16, 1988.
158. R.R. Murphy, *Introduction to AI Robotics*, MIT Press, Cambridge, MA, 2000.
159. R.M. Murray, Z. Li, S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, CA. 1994.
160. Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, Addison-Wesley, Reading, MA, 1991.
161. Y. Nakamura, H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.
162. Y. Nakamura, H. Hanafusa, "Optimal redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.
163. Y. Nakamura, H. Hanafusa, T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
164. J.I. Neimark, F.A. Fufaev, *Dynamics of Nonholonomic Systems*, American Mathematical Society, Providence, RI, 1972.
165. I. Nevins, D.E. Whitney, "The force vector assembler concept," *Proc. First CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Udine, I, 1973.
166. F. Nicolò, J. Katende, "A robust MRAC for industrial robots," *Proc. 2nd IASTED International Symposium on Robotics and Automation*, pp. 162–171, Lugano, Switzerland, 1983.
167. S. Nicosia, P. Tomei, "Model reference adaptive control algorithms for industrial robots," *Automatica*, vol. 20, pp. 635–644, 1984.

168. H. Nijmeijer, A. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, Berlin, Germany, 1990.
169. B. Noble, *Applied Linear Algebra*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1987.
170. C. O'Dúnlaing, C.K. Yap, "A retraction method for planning the motion of a disc," *Journal of Algorithms*, vol. 6, pp. 104–111, 1982.
171. K. Ogata, *Modern Control Engineering*, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.
172. D.E. Orin, R.B. McGhee, M. Vukobratović, G. Hartoch, "Kinematic and kinetic analysis of open-chain linkages utilizing Newton–Euler methods," *Mathematical Biosciences* vol. 43, pp. 107–130, 1979.
173. D.E. Orin, W.W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 66–75, 1984.
174. G. Oriolo, A. De Luca, M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 835–852, 2002.
175. R. Ortega, M.W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, pp. 877–888, 1989.
176. T. Patterson, H. Lipkin, "Duality of constrained elastic manipulation," *Proc. 1991 IEEE International Conference on Robotics and Automation*, pp. 2820–2825, Sacramento, CA, 1991.
177. T. Patterson, H. Lipkin, "Structure of robot compliance," *ASME Journal of Mechanical Design*, vol. 115, pp. 576–580, 1993.
178. R.P. Paul, *Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm*, memo. AIM 177, Stanford Artificial Intelligence Laboratory, 1972.
179. R.P. Paul, "Manipulator Cartesian path control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 702–711, 1979.
180. R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA, 1981.
181. R.P. Paul, B.E. Shimano, G. Mayer, "Kinematic control equations for simple manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 449–455, 1981.
182. R.P. Paul, H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *International Journal of Robotics Research*, vol. 5, no. 2, pp. 32–44, 1986.
183. D.L. Pieper, *The Kinematics of Manipulators Under Computer Control* memo. AIM 72, Stanford Artificial Intelligence Laboratory, 1968.
184. M.H. Raibert, J.J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 103, pp. 126–133, 1981.
185. E. Rimon, D.E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Proc. 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, pp. 21–26, 1989.
186. E.I. Rivin, *Mechanical Design of Robots*, McGraw-Hill, New York, 1987.
187. R.E. Roberson, R. Schwertassek, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin, Germany, 1988.
188. Z. Roth, B.W. Mooring, B. Ravani, "An overview of robot calibration," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 377–386, 1987.

189. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 2003.
190. J.K. Salisbury, "Active stiffness control of a manipulator in Cartesian coordinates," *Proc. 19th IEEE Conference on Decision and Control*, pp. 95–100, Albuquerque, NM, 1980.
191. J.K. Salisbury, J.J. Craig, "Articulated hands: Force control and kinematic issues," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.
192. C. Samson, "Robust control of a class of nonlinear systems and applications to robotics," *International Journal of Adaptive Control and Signal Processing*, vol. 1, pp. 49–68, 1987.
193. C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.
194. C. Samson, M. Le Borgne, B. Espiau, *Robot Control: The Task Function Approach*, Clarendon Press, Oxford, UK, 1991.
195. S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*, Springer-Verlag, Berlin, Germany, 1999.
196. V.D. Scheinman, *Design of a Computer Controlled Manipulator*, memo. AIM 92, Stanford Artificial Intelligence Laboratory, 1969.
197. J.T. Schwartz, M. Sharir, "On the 'piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, vol. 4, pp. 298–351, 1983.
198. L. Sciavicco, B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, pp. 550–559, 1986.
199. L. Sciavicco, B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 403–410, 1988.
200. L. Sciavicco, B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd ed., Springer, London, UK, 2000.
201. L. Sciavicco, B. Siciliano, L. Villani, "Lagrange and Newton–Euler dynamic modeling of a gear-driven rigid robot manipulator with inclusion of motor inertia effects," *Advanced Robotics*, vol. 10, pp. 317–334, 1996.
202. R. Sedgewick, *Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1988.
203. H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 472–490, 1989.
204. S.W. Shepperd, S.W., "Quaternion from rotation matrix," *AIAA Journal of Guidance and Control*, vol. 1, pp. 223–224, 1978.
205. R. Shoureshi, M.E. Momot, M.D. Roesler, "Robust control for manipulators with uncertain dynamics," *Automatica*, vol. 26, pp. 353–359, 1990.
206. B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, pp. 201–212, 1990.
207. B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint based robot control," *Robotica*, vol. 8, pp. 231–243, 1990.
208. B. Siciliano, J.-J.E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Proc. 5th International Conference on Advanced Robotics*, Pisa, I, pp. 1211–1216, 1991.

209. B. Siciliano, L. Villani, *Robot Force Control*, Kluwer, Boston, MA, 2000.
210. R. Siegwart, I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, MA, 2004.
211. D.B. Silver, "On the equivalence of Lagrangian and Newton–Euler dynamics for manipulators," *International Journal of Robotics Research*, vol. 1, no. 2, pp. 60–70, 1982.
212. J.-J.E. Slotine, "The robust control of robot manipulators," *International Jorunal of Robotics Research*, vol. 4, no. 2, pp. 49–64, 1985.
213. J.-J.E. Slotine, "Putting physics in control — The example of robotics," *IEEE Control Systems Magazine*, vol. 8, no. 6, pp. 12–18, 1988.
214. J.-J.E. Slotine, W. Li, "On the adaptive control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
215. J.-J.E. Slotine, W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
216. M.W. Spong, "On the robust control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1782–1786, 1992.
217. M.W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, Wiley, New York, 2006.
218. M.W. Spong, R. Ortega, R. Kelly, "Comments on "Adaptive manipulator control: A case study"," *IEEE Transactions on Automatic Control*, vol. 35, pp. 761–762, 1990.
219. M.W. Spong, M. Vidyasagar, "Robust linear compensator design for nonlinear robotic control," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 345–351, 1987.
220. SRI International, *Robot Design Handbook*, G.B. Andeen, (Ed.), McGraw-Hill, New York, 1988.
221. Y. Stepanenko, M. Vukobratović, "Dynamics of articulated open-chain active mechanisms," *Mathematical Biosciences*, vol. 28, pp. 137–170, 1976.
222. Y. Stepanenko, J. Yuan, "Robust adaptive control of a class of nonlinear mechanical systems with unbounded and fast varying uncertainties," *Automatica*, vol. 28, pp. 265–276, 1992.
223. S. Stramigioli, *Modeling and IPC Control of Interactive Mechanical Systems — A Coordinate Free Approach*, Springer, London, UK, 2001.
224. K.R. Symon, *Mechanics*, 3rd ed., Addison-Wesley, Reading, MA, 1971.
225. K. Takase, R. Paul, E. Berg, "A structured approach to robot programming and teaching," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 274–289, 1981.
226. M. Takegaki, S. Arimoto, "A new feedback method for dynamic control of manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 119–125, 1981.
227. T.-J. Tarn, A.K. Bejczy, X. Yun, Z. Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 114–122, 1991.
228. T.-J. Tarn, Y. Wu, N. Xi, A. Isidori, "Force regulation and contact transition control," *IEEE Control Systems Magazine*, vol. 16, no. 1, pp. 32–40, 1996.
229. R.H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM Journal of Research and Development*, vol. 23, pp. 424–436, 1979.
230. R.H. Taylor, D.D. Grossman, "An integrated robot system architecture," *Proceedings IEEE*, vol. 71, pp. 842–856, 1983.

231. S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.
232. L.W. Tsai, A.P. Morgan, "Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods," *ASME Journal of Mechanisms, Transmission, and Automation in Design*, vol. 107, pp. 189–200, 1985.
233. R. Tsai, "A versatile camera calibration technique for high accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Transactions on Robotics and Automation*, vol. 3, pp. 323–344, 1987.
234. J.J.Uicker, "Dynamic force analysis of spatial linkages," *ASME Journal of Applied Mechanics*, vol. 34, pp. 418–424, 1967.
235. L. Villani, C. Canudas de Wit, B. Brogliato, "An exponentially stable adaptive control for force and position tracking of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 44, pp. 798–802, 1999.
236. M. Vukobratović, "Dynamics of active articulated mechanisms and synthesis of artificial motion," *Mechanism and Machine Theory*, vol. 13, pp. 1–56, 1978.
237. M.W. Walker, D.E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 205–211, 1982.
238. C.W. Wampler, "Manipulator inverse kinematic solutions based on damped least-squares solutions," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, pp. 93–101, 1986.
239. L. Weiss, A. Sanderson, C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 404–417, 1987.
240. D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, pp. 47–53, 1969.
241. D.E. Whitney, "Force feedback control of manipulator fine motions," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 99, pp. 91–97, 1977.
242. D.E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 65–77, 1982.
243. D.E. Whitney, "Historical perspective and state of the art in robot force control," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 3–14, 1987.
244. W. Wilson, C. Hulls, G. Bell, "Relative end-effector control using Cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 684–696, 1996.
245. T. Yoshikawa, "Manipulability of robotic mechanisms," *International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
246. T. Yoshikawa, "Dynamic manipulability ellipsoid of robot manipulators," *Journal of Robotic Systems*, vol. 2, pp. 113–124, 1985.
247. T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators — Description of hand constraints and calculation of joint driving force," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 386–392, 1987.
248. T. Yoshikawa, *Foundations of Robotics*, MIT Press, Boston, MA, 1990.
249. T. Yoshikawa, T. Sugie, N. Tanaka, "Dynamic hybrid position/force control of robot manipulators — Controller design and experiment," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 699–705, 1988.

250. J.S.-C. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 434–440, 1988.

Index