Universidade Federal de Santa Catarina
Programa de Pós-Graduação em Engenharia Mecânica

Análise Cinemática Hierárquica de Robôs Manipuladores

Tese de Doutoramento submetida à
Universidade Federal de Santa Catarina
como requisito parcial à obtenção do grau de

Doutor em Engenharia Mecânica

por

Daniel Martins

Florianópolis, Fevereiro de 2002

À minha esposa
Andréia
e às minhas filhas
Barbara Augusta e Maria Eugênia...

Agradecimentos

Ao Prof Raul Guenther, pela amizade e pela paciência em me guiar pela árdua tarefa da escrita da tese.

Aos colegas Henrique Simas, Alexandre Campos, Emerson Raposo, Júlio Feller Golin, Waldoir Valentim Gomes Junior, Lucas Weihmann, Carlos Henrique Santos

Ao Prof José Carlos Zanini, que me iniciou nas lides da teoria de mecanismos e máquinas.

Aos meus alunos e orientados, que muito me serviram de inspiração e motivação para a execução deste trabalho.

Acknowledgements

*To Dr Craig Tischler and Prof Kenneth Hunt (*in memoriam*) for their patience in revealing me the inner secrets of screw theory and geometry.*

To Dr David Downing and Dr Stuart Lucas for a pleasant company and very helpful comments when I was in Melbourne and even after my short visit to the University of Melbourne.

Contents

List of Figures p. ii

List of Tables p. iii

Abstract p. iv

Abstract p. v

1 Introduction p. 1

1.1 Singularities . . . . . p. 2

1.2 Purposes of the method . . . . . p. 3

1.3 Hierarchical analysis: a simple example . . . . . p. 4

1.4 Extensions to the method . . . . . p. 7

1.5 Overview of the work . . . . . p. 8

2 Hierarchical Kinematic Analysis p. 10

2.1 Permutation matrices . . . . . p. 10

2.2 Digraph Analysis of Jacobian Matrices . . . . . p. 12

2.2.1 Graph definitions . . . . . p. 13

2.2.2 Strong components of a digraph . . . . . p. 15

2.2.3 Related matrices $\bar{R}, \bar{S}, \bar{T}$ . . . . . p. 18

2.3 Hierarchical Jacobian reordering . . . . . p. 23

3 Jacobian Matrices and Screws p. 25

3.1 Sparsity of the Jacobian matrix . . . . . p. 25

3.2 Denavit-Hartenberg method . . . . . p. 26

3.3 Velocities along a serial chain . . . . . p. 27

3.4 Screw-based Jacobian matrix . . . . . p. 31

3.5 Screw dependence of the joint variables . . . . . p. 34

3.6 Line permutations of the Jacobian matrix . . . . . p. 39

4 Serial Robots p. 41

4.1 Hierarchical analysis of a Puma robot . . . . . p. 41

4.2 Hierarchical canonical form . . . . . p. 45

4.3 Jacobian Inversion . . . . . p. 46

4.4 Hierarchical analysis . . . . . p. 47

4.5 Other advantages . . . . . p. 49

5 Redundant Robots p. 51

5.1 Introduction to redundant robots . . . . . p. 51

5.2 Redundancy resolution . . . . . p. 53

5.3 Extended Jacobian method . . . . . p. 54

5.3.1 Comparison with generalised inverse methods . . . . . p. 56

5.3.2 Hierarchical kinematic analysis using extended Jacobian method . . . . . p. 58

5.4 Globally constrained Jacobian method . . . . . p. 58

5.4.1 The simplest choice $\vec{n} = e_i$ . . . . . p. 60

5.4.2 Algorithmic singularities . . . . . p. 61

5.4.3 Hierarchical kinematic analysis using globally constrained Jacobian method . . . . . p. 62

5.5 Hierarchy of the singularities of the Roboturb robot . . . . . p. 62

5.6 Final comments . . . . . p. 68

6 Parallel Robots p. 69

6.1 Overview of the chapter . . . . . p. 70

6.2 Requisites of the hierarchical kinematic analysis . . . . . p. 70

6.3 Parallel robots: an introduction . . . . . p. 71

6.3.1 Stewart-Gough platforms . . . . . p. 73

6.4 Kinematic problem of parallel robots . . . . . p. 74

6.5 Kirchhoff-Davies laws and parallel robots . . . . . p. 75

6.6 Hierarchical kinematic analysis of parallel robots . . . . . p. 79

6.7 Permutation of the system . . . . . p. 81

6.7.1 Actuators at joints A, F, G . . . . . p. 82

6.7.2 Actuators at joints D, E, F . . . . . p. 83

6.7.3 Actuators at joints A, F, E . . . . . p. 84

6.8 Assur groups . . . . . p. 86

6.9 Conclusions about parallel robots . . . . . p. 88

7 Conclusions p. 90

7.1 Perspectives and further work . . . . . p. 91

Appendix A – Screw Theory p. 94

A.1 Kinematics and Statics of a rigid body . . . . . p. 94

A.1.1 Line vectors and free vectors . . . . . p. 95

A.1.2 Statics . . . . . p. 96

A.1.3 Kinematics . . . . . p. 97

A.2 Screw Theory . . . . . p. 98

A.2.1 Twists × Wrenches . . . . . p. 99

A.2.1.1 Twists . . . . . p. 99

A.2.1.2 Wrenches . . . . . p. 100

A.2.2 Screws . . . . . p. 100

A.2.3 Plücker coordinates . . . . . p. 104

A.2.3.1 Plücker line coordinates . . . . . p. 104

A.2.3.2 Plücker screw coordinates . . . . . p. 105

A.2.4 Normalised screws . . . . . p. 107

A.2.5 Ray order × Axis order . . . . . p. 110

A.2.6 Virtual Work . . . . . p. 113

A.3 Screw Coordinate Transformation . . . . . p. 117

A.3.1 Ray-axis screw coordinate transformation . . . . . p. 118

A.3.2 Translation of axes . . . . . p. 118

A.3.3 Rotation of axes . . . . . p. 120

Appendix B – Chasles' and Poinsot's lines p. 123

B.1 Screw independence on representation . . . . . p. 123

B.1.1 Pitch invariance . . . . . p. 124

B.1.2 Axis invariance . . . . . p. 124

B.2 A slightly different approach . . . . . p. 125

Appendix C – Matrix powers p. 127

Appendix D – Numerical Inverse Jacobian p. 130

Appendix E – Kirchhoff-Davies circuit law p. 133

E.1 Adaptation of the Kirchhoff laws to Kinematics . . . . . p. 133

E.2 Coupling graph of a kinematic chain . . . . . p. 134

E.3 Movement graph of a kinematic chain . . . . . p. 137

E.4 Kirchhoff-Davies circuit law . . . . . p. 139

References p. 143

List of Figures

1 The SCARA robot. . . . . p. 5

2 Directed graph (digraph) . . . . . p. 13

3 Matrix, incidency matrix and digraph . . . . . p. 13

4 Digraph of the Scara Jacobian according to eq. (1.2) . . . . . p. 16

5 Condensation of the digraph related to the Scara Jacobian . . . p. 17

6 Manipulator with all joints frozen . . . . . p. 28

7 manipulator with all but the last joint frozen . . . . . p. 29

8 manipulator with all but the penultimate joint frozen . . . . . p. 30

9 Representation of a normalised screw . . . . . p. 32

10 Sequence of links in a serial kinematic chain with the coordinate system fixed at link i. . . . . p. 36

11 Sequence of links in a serial kinematic chain with the coordinate system fixed at link i making explicit the screw dependence of the joint variables $\dot{q}_i$. Mobile links are filled in black. . . . . p. 36

12 The Puma robot analyzed in this work. The screw $\$_5$, which is not explicitly designated on the Figure is on the joint axis 5 in the middle of the spherical wrist. . . . . p. 42

13 Side view (xz-plane) of the Puma. . . . . p. 42

14 Frontal view (yz-plane) of the Puma highlighting the spherical wrist. . . . . p. 43

15 A sketch of the Puma robot with all joint variables at initial position.The coordinate system highlighted has origin at the wrist centre and is fixed on link 3. . . . . p. 44

16 Condensation digraph of a Puma robot Jacobian based on the system given by eq. (4.3). Vertex A of the condensation digraph is the source of the digraph and vertex E is the sink of the digraph. A singularity at the vertex D divides the condensation digraph in two parts: one affected by the singularity and the other free from singularity effects . . . . . p. 49

17 Comparison of the constraints of the globally constraint Jacobian method and the classical extended Jacobian method seen from a 2-dimensional projection of the $\mathbb{R}^7$ space. Scalars $\bar{r}_i$ are projections of the values of r in eq. (5.8). Negative values for r, and correspondingly $\bar{r}_i$ are also possible. . . . . p. 60

18 The Roboturb manipulator . . . . . p. 63

19 Coordinate system chosen to the Roboturb robot . . . . . p. 64

20 The planar Stewart platform $3RRR$ being actuated at joints A, F, G. The greater triangle 1 is attached to the base (fixed link or fixed platform) and the triangle 4 is the moving link or moving platform . . . . . p. 75

21 Details of the the dimensions (lengths and angles) as well as of the chosen coordinate system to represent the mechanism $3RRR$ in the xy-plane . . . . . p. 76

22 Coupling graph of mechanism $3RRR$. . . . . p. 76

23 Chords and circuits of the movement graph G_M for the mechanism $3RRR$. This graph coincides with the coupling graph G_C as there is no kinematic pairs with $f > 1$. . . . . p. 77

24 Mechanism $3RRR$ actuated at joints A, F, G . . . . . p. 82

25 Mechanism $3RRR$ actuated at joints D, E, F . . . . . p. 83

26 Mechanism $3RRR$ actuated at joints A, F, E . . . . . p. 85

27 Pair of first order Assur groups derived from mechanism $3RRR$ actuated at joints D, E, F shown in Fig. 25. The pair of Assur groups, in this case two dyads, can be solved in any order, *e.g.* Assur group 1 can be solved after or before Assur group 2. . p. 88

28 Pair of first order Assur groups derived from mechanism $3RRR$ actuated at joints A, F, E shown in Fig. 26. Notice that this Figure as a whole is non-minimal Assur group, but it can be split up into two minimal Assur groups, in this case two dyads which must be solved in order: Assur group 1 *always* before Assur group 2. . . . . p. 89

29 Second order Assur group derived from the mechanism $3RRR$ actuated at joints A, F, G shown in Fig. 24. Notice that this Figure as a whole is a minimal Assur group, so it cannot be split up into smaller Assur groups. . . . . p. 89

30 Line-vectors a) Forces b) Angular velocities. . . . . p. 96

31 Chasles' line and Poinsot's line. . . . . p. 101

32 Twist represented by a screw (plus a dimensional magnitude). The velocity of a point A is given by $v_A = \tau + \omega \times \vec{OA}$ . . . . . p. 111

33 Wrench represented by a screw (plus a dimensional magnitude). The summation of couples at a point A is given by $M_A = C + F \times \vec{OA}$ . . . . . p. 111

34 A wrench $\$^w$ acting on a twist $\$^t$. . . . . p. 114

35 Virtual work of a wrench acting on a twist. . . . . p. 115

36 Details of the point dependent vector decomposition relative to the wrench and to the twist. . . . . p. 115

37 Mechanism $3RRR$ in the xy-plane . . . . . p. 135

38 Graph of mechanism $3RRR$. . . . . p. 135

39 Chords and circuits of the graph G_C for the mechanism $3RRR$. p. 137

40 Chords and circuits of the graph G_M for the mechanism $3RRR$ p. 138

List of Tables

1 Normalised screws $\hat{\$}$, pitches h and magnitudes q_i for rotative and prismatic pairs . . . . . p. 32

2 Comparison between Statics and Kinematics. All variables are represented in SI units and are taken relative to a generic point P p. 110

3 Combination of factors to generate the virtual work . . . . . p. 116

Abstract

Devido a sua estrutura cinemática, os robôs manipuladores seriais apresentam singularidades, que são definidas como as configurações onde a mobilidade do manipulador é reduzida. Matematicamente estas configurações correspondem àquelas onde a matriz Jacobiana perde posto. As singularidades introduzem problemas nas operações do robô que precisam ser evitadas.

Normalmente isto é feito detectando-se *a priori* a configuração singular e então desviando-se dela. Os métodos usuais para o evitamento de singularidades consideram implicitamente todas as singularidades igualmente desastrosas à cinemática do robô. Este fato torna mais difícil o estabelecimento de uma estratégia para o desvio de singularidades.

Esta tese introduz um método novo de análise, designado de *análise cinemática hierárquica*, que é baseado no fato de que as singularidades cinemáticas do robô *não* são igualmente importantes para o comportamento do manipulador. As singularidades são classificadas de acordo com a sua influência na cinemática inversa do robô. Além disso, este método fornece um algoritmo recursivo para resolver mais rapidamente, e com maior exatidão, o problema inverso da cinemática de robôs seriais. O método, que é baseado em uma interpretação gráfica da matriz Jacobiana, é sumarizado em um algoritmo.

Para ilustrar sua utilidade, este algoritmo é aplicado a um robô Puma. Os resultados são interpretados e, mesmo neste exemplo clássico, alguns fatos novos podem ser extraídos usando a análise cinemática hierárquica.

Nos últimos capítulos o método é estendido a dois casos mais gerais: robôs redundantes e robôs paralelos. A solução de robôs redundantes requereu uma nova formulação para resolver a redundância. A solução de robô paralelos não utiliza diretamente a matriz Jacobiana mas sim uma matriz auxiliar de solução. O procedimento requereu o uso das leis de Kirchhoff-Davies e os resultados apresentam uma conexão direta com o conceito clássico de grupos de Assur.

Abstract

Due to their kinematic structure, serial robot manipulators present singularities, which are defined as the configurations where the mobility of the manipulator is reduced. Mathematically, these singular configurations are those where the Jacobian matrix of the robot drops rank. Singularities introduce problems in the robot operations that need to be avoided.

Usually this is done by detecting the singular configuration and then deviating from them. The usual methods for singularity avoidance implicitly consider all singularities equally disastrous to the robot kinematics; therefore, the establishment of an strategy to deviate from the singularities becomes more difficult.

This paper introduces a new analysis method, designated as the hierarchical kinematic analysis, which is based on the fact that the robot kinematic singularities are *not* equally important to the behaviour of the manipulator. Singularities are ranked according to the their influence on the robot inverse kinematics. Moreover, this method provides a recursive algorithm to solve faster, and with more accuracy, the inverse kinematics problem of serial robots. The method, which is based on a graphical interpretation of the Jacobian matrix, is summarized in a algorithm.

To illustrates its usefulness, this algorithm is applied to a Puma robot. The results are interpretated and, even in this classical example, some new facts can be extracted by using the hierarchical kinematic analysis.

In the last chapters, the method is extended to two more general cases: redundant robots and parallel robots. The redundant robot solution demanded for a different form of solving the redundancy. The parallel robot solution does not use the Jacobian matrix directly, but an intermediary matrix. The procedure was based on the Kirchhoff-Davies laws, and the results have a close connection with the classical concept of Assur groups.

1 Introduction

Mas eu que falo, humilde, baxo e rudo,
De vós não conhecido nem sonhado?
Da boca dos pequenos sei, contudo,
Que o louvor sai às vezes acabado.
Tem me falta na vida honesto estudo,
Com longa experiência misturado,
Nem engenho, que aqui vereis presente,
Cousas que juntas se acham raramente.

Luís de Camões – Os Lusíadas, Canto Décimo – 154

As a rule, the end-effector of a serial robot is programmed to follow a set of desired positions and orientations in the Cartesian space. Positions and orientations first derivatives (velocities) are often imposed as extra conditions on the path tracking. Transforming these motion specifications into the corresponding joint space motions requires solving the inverse kinematics problem, which is the determination of the joint variables related to a given end-effector position and orientation.

The inverse kinematics, at differential level, can be represented by a linear system

$$J(q)\dot{q} = \dot{x} \tag{1.1}$$

where vector q comprises the joint variables, *e.g.* rotation angles if all joint are rotative. Vector $\dot{q}$ of joint velocities, the time derivative of vector q, is related

to vector $\dot{x}$ of end-effector velocities, specified as the desired velocities, by the Jacobian matrix $J(q)$. The matrix $J(q)$, also designated simply by Jacobian, must be inverted to solve eq. (1.1).

The inverse of the Jacobian, J^{-1}, may be used to solve the inverse kinematic problem even when only positions and orientations (and not their derivatives) are specified. Inverting the Jacobian avoids solving the inverse kinematic problem in closed form, as closed-form solutions are available only to simple kinematic structures [Sciavicco e Siciliano 1996].

So, in general, the inverse kinematic problem solution demands the inversion of the Jacobian matrix, a computationally expensive route [Lucas, Tischler e Samuel 2000], particularly when this inversion has to be done online [Zomaya, Smith e Olariu 1999].

1.1 Singularities

The Jacobian is a function of the robot geometrical parameters and, normally, of the configuration q; therefore, it may drop rank at some configurations. Those configurations q where $J(q)$ is rank deficient are termed kinematic singularities. Kinematic singularities, in serial robots, represent configurations where the mobility of the structure is reduced, *i.e.* an arbitrary movement to the end-effector cannot be imposed. Furthermore, small velocities in the Cartesian space may require large velocities in the joint space when the manipulator reaches the neighbourhood of a singularity.

The problems introduced by singularities need to be avoided, *e.g.* by deviating from those singular configurations. Avoiding singularities is normally done off-line, *i.e.* first these configurations are to be *a priori* detected and then, when the robot is moving and before the singularity is reached, they are deviated from. In the worst case, when a singularity is reached, another strategy has to be employed to escape from the singularity within feasible joint velocities.

A matrix J drops rank when *e.g.* two columns (or rows) are linearly dependent each other. In this case $\det J = 0$. The matrix J also drops rank when a line (row or column) is null; therefore, $\det J = 0$ is also satisfied in this case. So, an equation like $\det J = 0$ cannot be used to distinguish among these quite different cases. The use of this equation to detect singularities is possibly originated by the widely spread idea that all singularities are *equally* disastrous to the manipulator kinematics, as each singularity leads $\det J$ to zero "equally".

Standard algorithms, see a review at [Nakamura e Hanafusa 1986], detect the proximity of a singularity but do not specify, or simply do not care about, *which particular* singularity is being reached. Such algorithms, most of them based on the determinant or the singular values of the Jacobian, turn the differences among singularities hardly discernible.

1.2 Purposes of the method

By introducing a new analysis method, we intend to achieve three main purposes. The first one is to provide a recursive algorithm to solve faster, and with more accuracy, the inverse kinematics problem of serial robots based on a preliminary Jacobian manipulation. The second purpose is to outline that all singularities are *not* equally important to the behaviour of the manipulator. Finally, the third purpose is to explicitly determine the set of variables affected by and responsible for each singularity.

We designate this new method as the *hierarchical kinematic analysis*, which is a method based entirely on the structure of the Jacobian and on the reduction of the Jacobian matrix to its finest block-triangular form. Sometimes this reduction may be done easily by inspection, as described in section 1.3 for the SCARA robot, but this only applies to quite simple manipulators. Moreover, the block-triangular form obtained by inspection cannot be ensured to be the finest one, *e.g.* another rearrangement, not so obvious, may yield smaller diagonal blocks. In other words, a systematic method which leads a sparse matrix to its finest block-

triangular form is needed. Such method, based on directed graphs, is described in chapters 2-4.

The Jacobian matrix sparsity is crucial to all methods which represent matrices by graphs [Pissanetsky 1984]. Chapter 3 describes how screw theory can be employed to enhance the sparsity of the Jacobian matrix before using the method described in chapter 2

A complete hierarchical kinematic analysis is performed on a Puma robot in chapter 4.

1.3 Hierarchical analysis: a simple example

The usefulness of the proposed hierarchical analysis will be outlined by a simple example; therefore, we intentionally chose a SCARA robot with four degrees of freedom $(\theta_1, \theta_2, d_3, \phi)$ along three axes, Fig. 1. The Jacobian matrix of this robot, relating the velocity vector in the Cartesian space, $\dot{x}_{scara} = [v_x\ v_y\ v_z\ \omega_z]^T$, and the joint velocity vector, $\dot{q}_{scara} = [\dot{\theta}_1\ \dot{\theta}_2\ \dot{d}_3\ \dot{\phi}]^T$, may be described by [Spong e Vidyasagar 1989]

$$\dot{x} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_z \end{bmatrix} = \begin{bmatrix} y_{12} & -a_2 s_{12} & 0 & 0 \\ x_{12} & a_2 c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\phi} \end{bmatrix} = J\dot{q} \tag{1.2}$$

where a_i is the length of link i; $y_{12} = -a_1 s_1 + a_2 s_{12}$; $x_{12} = -a_1 c_1 + a_2 c_{12}$; $\dot{\theta}_1, \dot{\theta}_2, \dot{\phi}$ are the rate of rotation about the revolute joints; $\dot{d}_3$ is the rate of the linear displacement of the prismatic joint; and s_i, c_i, s_{12}, c_{12} are the commonly accepted short notation for $\sin\theta_i$, $\cos\theta_i$, $\sin(\theta_1 + \theta_2)$, $\cos(\theta_1 + \theta_2)$, respectively.

The hierarchical analysis is based on reordering the Jacobian into an upper block-triangular form. The finest reordering scheme for this simple example almost reveals itself and it can be obtained following two steps: first changing the rows upside-down, and then reordering the columns.

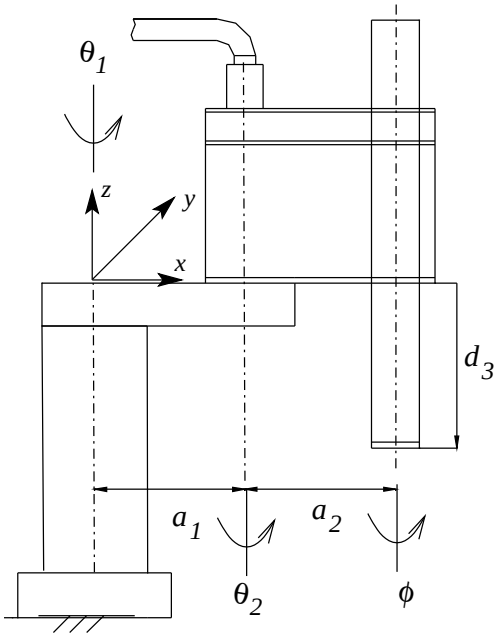

Figure 1: The SCARA robot.

With the rows upside-down, *i.e.* row order 4321, eq. (1.2) becomes

$$\begin{bmatrix} \omega_z \\ v_z \\ v_y \\ v_x \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ x_{12} & a_2c_{12} & 0 & 0 \\ y_{12} & -a_2s_{12} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\phi} \end{bmatrix} \tag{1.3}$$

Reordering again the system (1.3), this time columns in order 4312, the SCARA Jacobian becomes

$$\check{\dot{x}} = \begin{bmatrix} \omega_z \\ v_z \\ v_y \\ v_x \end{bmatrix} = \check{J}_{\text{scara}}\check{\dot{q}} = \left[\begin{array}{c|c|cc} -1 & 0 & 1 & 1 \\ \hline 0 & -1 & 0 & 0 \\ \hline 0 & 0 & x_{12} & a_2c_{12} \\ 0 & 0 & y_{12} & -a_2s_{12} \end{array}\right] \begin{bmatrix} \dot{\phi} \\ \dot{d}_3 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{1.4}$$

where the reordered Jacobian, $\check{J}_{\text{scara}}$, has the desired upper block-triangular form.

The upper block-triangular form of eq. (1.4) turns the inversion of the Jacobian matrix easier. In fact, the inversion can be done systematically by an

(almost) back-substitution process. From bottom to top the whole system (1.2) becomes a chained series of subsystems to be solved in order. For example, the solution of the inverse kinematic problem is obtained by the following steps

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} x_{12} & a_2 c_{12} \\ y_{12} & -a_2 s_{12} \end{bmatrix}^{-1} \begin{bmatrix} v_y \\ v_x \end{bmatrix} \tag{1.5}$$

$$\dot{d}_3 = -v_z \tag{1.6}$$

$$\dot{\phi} = -1 \cdot \omega_z + \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{1.7}$$

where eq. (1.7) is solved only after obtaining the values of eq. (1.5). Equation (1.5), in turn, requires the inversion of a simple 2×2 system. Briefly, this process uses less memory, induces fewer errors, and is easier to compute.

The block-triangular form, eq. (1.4), allows establishing a hierarchy of variables, because the relationship among variables becomes apparent. The block-triangular form itself describes how the variables affect one another and, maybe more importantly, which group of variables do not affect some other group of variables.

For instance, $\dot{d}_3$ exclusively depends upon v_z and vice-versa. Similarly, $\dot{\theta}_1, \dot{\theta}_2$ only depend upon v_y, v_x and vice-versa. Moreover $\dot{\phi} = \dot{\phi}(\omega_z; \dot{\theta}_1, \dot{\theta}_2)$ and $\omega_z = \omega_z(\dot{\phi}; v_y, v_x)$.

Furthermore, the upper block-triangular form of eq. (1.4) introduces other advantages when compared to the original form (1.2). Some of these advantages appears even in this simple example, like:

- the hierarchy of diagonal blocks highlights a hierarchy of singularities. Unfortunately, this fact cannot be inferred from eq. (1.4) as the first diagonal blocks are scalar and do not vary with the movement of the robot, so there is only one singularity for the SCARA robot. This problem will be discussed for a Puma robot in chapter 4.

- the pivoting process in a numerical LU decomposition [Golub e Loan 1983]

of an upper block-triangular matrix is almost done without aggregating errors to the solution. Solving directly eq. (1.4) is better than solving eq. (1.2) even when the decomposition, like eqs. (1.5)- (1.7), is not used.

- the determinant is easier to obtain. From eq. (1.4) we observe that the determinant is explicitly constrained to the determinant of the 2×2 diagonal block at the bottom of system: $\begin{bmatrix} x_{12} & a_2c_{12} \\ y_{12} & -a_2s_{12} \end{bmatrix}$.

1.4 Extensions to the method

Above, we described the hierarchical kinematic analysis from a nonredundant serial robot point of view. However, this method can be extended to other types of manipulators, namely redundant and parallel robots. These extensions requires some adaptation and will be described in chapter 5, for redundant robots, and chapter 6, for parallel robots.

Redundant robots, chapter 5, have non-square Jacobian matrices, turning difficult the inversion and the use of the hierarchical kinematic analysis. One approach to the redundancy resolution is the *extended Jacobian method*, *i.e.* the addition of an extra line to "square" the Jacobian; the hierarchical kinematic analysis could be, in principle, applied to this extended Jacobian.

The problem in the classical extended Jacobian methods is the selection of the extra row. These classical methods rely on general minimisation criteria and these criteria normally couple all the variables that were not originally coupled. To overcome this difficulty we propose an alternative variant of the extended Jacobian method, called *globally constrained Jacobian method*. This variant has advantages over the extended Jacobian method in the setting of the constraints and in the singularity avoidance.

Parallel robots, chapter 6, have an even higher complexity in their kinematics. Normally the inverse kinematics problem for parallel robots is considerably simpler than the direct kinematics problem. The closure of the kinematic chain

impose a first, and quite important, problem: the determination of the parameters of all joints.

Differently from serial and redundant robots, not all joints in a parallel robot are actuated. Some of the joints in the closed kinematic chain of a parallel robot are passive, and their parameters (position, velocity, acceleration, ...) are dependent upon the actuated joints parameters. The calculation of these parameters at differential level (velocities) can be posed as a linear system obtained by the Kirchhoff-Davies laws, see section 6.5 at page 75 and appendix E at page 133.

The solution of this linear system may be improved by the hierarchical kinematic analysis. As a collateral effect, we notice a close relationship between the block-triangular form of the rearranged system and the intrinsic kinematic structure of the mechanism, in this case represented by Assur groups, section 6.8 at page 86.

1.5 Overview of the work

Chapter 2, gives the graph theoretical basis of the hierarchical kinematic analysis. This chapter relates matrices and graphs and discusses how the sparsity of a matrix can be represented by its associated graph. The main result of the chapter is the Algorithm 1 at page 22 that leads a matrix to its finest block-triangular form by independent permutations of rows and columns.

Chapter 3 shows the relationship between the sparsity of the Jacobian and the geometry of the robot. It discusses also the choice of the "best" coordinate system, *i.e.* the system which leads to a sparser Jacobian.

Chapter 4 details the hierarchical kinematic analysis to serial robots. The hierarchical kinematic analysis is applied to the Puma robot, and even in this classical example some new results appear.

Chapter 5 extends the hierarchical kinematic analysis to redundant robots. It also discusses some methods to the redundancy resolution, in special the extended

Jacobian method. In this chapter a variant of the extended Jacobian method is proposed: the globally constrained Jacobian method. The results are applied to the Roboturb manipulator. The Roboturb manipulator is a redundant welding manipulator, designed and built at the Universidade Federal de Santa Catarina in a joint venture with the LacTec institute (Curitiba-Brazil), for repairing eroded turbine blades.

Chapter 6 extends the hierarchical kinematic analysis to parallel robots. This chapter also discusses some prerequisites of the hierarchical kinematic analysis which were trivially satisfied by the serial and redundant formulations. The proposal of the chapter uses the Kirchhoff-Davies laws to obtain the parallel robot kinematics. The results are applied to a planar version of the Stewart-Gough platform, the 3RRR manipulator.

Appendix A develops screw theory from a dual vector approach. Contrarily to most texts in screw theory, this chapter describes some fundamentals of rigid body motion with no pre-requisite. The duality statics-kinematics is also presented. The text do not use advanced geometrical concepts such as projective geometry and Plücker's conoid (cylindroid). They are mentioned only when necessary. No previous knowledge of these topics is required.

Appendix B demonstrates the theorems of Chasles and Poinsot, cite and used in appendix A, in two different ways. This appendix also details some properties, as the screw representation invariance, section B.1.

Appendix C discusses the matrix powers in the context of graph analysis. It completes the demonstration of some steps before the Algorithm 1.

Appendix D provides an alternative way of obtaining the structure of the inverse matrix used in Algorithm 1. The structure of the inverse matrix is used without applying symbolic inversion.

Appendix E details the Kirchhoff-Davies laws used in the parallel robot kinematic analysis.

2 *Hierarchical Kinematic Analysis*

2.1 Permutation matrices

The tougher step in the whole process described in section 1.3 is to obtain the order of rows and columns that reduce the Jacobian to an upper block-triangular form. The row order 4321 and column order 4312 are selected from the $4! \times 4! = 576$ possible independent permutations of rows and columns. For the Scara robot these permutations can be found by inspection or, at most, with a little number of trials.

An equivalent way to look at the search of a row order, like 4321 from eq. (1.2) to eq. (1.3), and a column order, like 4321 from eq. (1.3) to eq. (1.4), is to look for their corresponding permutation matrices. Hence, the problem can be restated as: to find a pair of permutation matrices, P_r for rows and P_c for columns, that reorder the system described in eq. (1.2).

A permutation matrix is a matrix whose rows are formed by versors e_i where the only non-null element is the i-th element which has value 1. For instance, in the example above, the process to obtain $\check{\dot{x}}$ from $\dot{x}$ can be described by the

following procedure

$$\check{\dot{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \dot{x} = \begin{bmatrix} \cdots & \vec{e}_4 & \cdots \\ \cdots & \vec{e}_3 & \cdots \\ \cdots & \vec{e}_2 & \cdots \\ \cdots & \vec{e}_1 & \cdots \end{bmatrix} \dot{x} = P_r \dot{x} \tag{2.1}$$

Permutation matrices have interesting properties. An important property is that permutation matrices P are orthonormal matrices and so $PP^T = P^T P = I$. Using this property and eq. (2.1) we can obtain eq. (1.4) from eq. (1.2) by a different route. Defining $\check{\dot{x}} \triangleq P_r \dot{x}$, the rearranged Cartesian velocities vector, and $\check{\dot{q}} \triangleq P_c \dot{q}$, the rearranged joint velocities vector, we obtain

$$\check{\dot{x}} = \check{J} \check{\dot{q}} \tag{2.2}$$

where $\check{J} = P_r J P_c^T$ is the reordered Jacobian matrix. The permutation matrix P_r corresponds to the row permutation of the J and the permutation matrix P_c corresponds to the column permutation of the J.

With this new definitions, the problem of reordering the Jacobian matrix can now be restated as: find a pair of permutation matrices, P_r for rows and P_c for columns, such that the reordered Jacobian matrix $\check{J} = P_r J P_c^T$ has the *finest* upper block-triangular form. The finest upper block-triangular form of the Jacobian matrix will be designated as the *hierarchical canonical form* of the Jacobian.

Another question that arises is whether the obtained block-triangular form can be reordered to a finer upper block-triangular form *i.e.* with smaller diagonal blocks. Eq. (1.4) has only one 2×2 diagonal block and, again by inspection, we can notice that this block cannot be split into two one-dimensional blocks by any permutation. Any attempt to split this block will unavoidably create another 2×2 diagonal block or destroy the block-triangular form.

The hierarchical canonical form for more complex robot architectures, such as those with 6 degrees of freedom, will hardly expose itself so easily as the block-

triangular form of the Scara robot. So there is a need of a systematic search method for this form. Next sections will describe an algorithmic method, based on directed graphs, which obtains the hierarchical canonical form of the Jacobian matrix.

2.2 Digraph Analysis of Jacobian Matrices

Above we were acquainted with the necessity of a systematic method to derive the block-triangular form of a sparse matrix. Several methods exist to reduce a matrix to a block-triangular form. Most of these methods are based on graphs [Pissanetsky 1984], yet there are some works which propose a matroid approach [Murota 2000]. Depending of the type of the matrix and computational requirements a certain type of graph is required.

Pissanetsky (1984) makes a good review of the graph-based methods from the computational point of view. If the matrix is symmetrical the most natural representation is based on non-directed graphs; however, if the matrix is general, *i.e.* not symmetrical, two options appears: bipartite graphs and directed graphs. Bipartite graphs implicitly apply rows and columns permutations to the matrix, while directed graphs are restricted to the simultaneous row and column permutation. On the other hand, we believe that directed graphs expresses more clearly the interdependency of the variables and, therefore, clarify the flow of variables from one step to another. In this work, we obtain both independent permutations on the Jacobian matrix and, at the same time, these rearrangements are applied to the inverse Jacobian, by introducing the kinematic structure matrix in section 2.3.

Now, we start describing a method based on directed graphs; the main concepts will be summarized in this and in the following sections. More detail on this topic can be found on [Murota 2000] [Harary, Norman e Cartwright 1965] [Reinschke 1988] and references therein.

2.2.1 Graph definitions

A *graph* G is a structure which consist of *vertices* or nodes and links. The links can be orientated, *arcs*, or non-orientated,*edges* A vertex is a point in the graph and an edge is a line connecting two vertices. A *subgraph* of a graph G is a graph whose vertices and links are vertices and links of G.

Some graph representations assign an orientation to each link of the graph. This modified graph is called a *directed graph*, or *digraph* for short, and the orientated edges are called arcs. Each arc a_{kj} in a digraph is directed from an initial vertex j to an end vertex k (Fig. 2)[1]. The graph representations used in this work are based uniquely on digraphs.

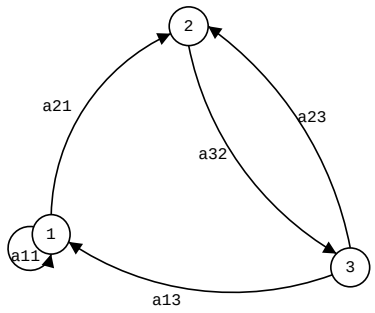

Figure 2: Directed graph (digraph)

$$\bar{A} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & 0 & a_{13} \\ a_{21} & 0 & a_{23} \\ 0 & a_{32} & 0 \end{bmatrix}$$

Figure 3: **Left:** incidency matrix $\bar{A}$ of digraph in Fig. 2. **Right:** matrix A with variables elements which has incidency matrix $\bar{A}$ and can be represented by the directed graph in Fig. 2

We can associate each digraph to an incidency matrix. The incidency matrix[2] $\bar{A} = [a_{ji}]$ is a matrix whose elements are defined by

$$a_{ji} \triangleq \begin{cases} 1 & \text{if there is an arc from vertex } i \text{ to vertex } j \\ 0 & \text{if, otherwise, there is no such arc.} \end{cases} \tag{2.3}$$

[1] It may appear weird the apparent "inversion" of indices. The more natural association – arc a_{kj} with initial vertex k and end vertex j – will not be used in this text to keep coherence with the references [Reinschke 1988] and [Jantzen 1996]. These books use the "inverted" notation which is more convenient in control problems. As we followed the methodology described in these works and we intend to extend in the future the work beyond the robot kinematics, *e.g.* to the robot control problem, we will keep the "inverted" notation along the remaining of the work. A consequence of this notation is the concept of adjacency matrix instead of the classical concept of incidency matrix.

[2] Again to follow the notation of [Reinschke 1988] and [Jantzen 1996] we use the definition of incidency matrix instead of the adjacency matrix. Other books on graph theory define incidency matrix as the matrix which associates the edges or arcs and the circuits.

The incidency matrix is a Boolean matrix in a sense that their elements, only 0's or 1's, satisfy Boolean arithmetics operations. Boolean arithmetics share most of the properties of the real numbers. The only difference between Boolean addition and real addition is $1+1 = 1$. At first glance, it seems to be an awkward result but, in fact, it is derived from the logical equation $1 \vee 1 = 1$ *or* $1 = 1$. Boolean matrices are here designated by Roman capital letters with an overbar like $\bar{A}, \bar{R}, \bar{T}, \ldots$ A graph related to an incidency matrix $\bar{A}$ is designated by $G(\bar{A})$.

A specific labelling of vertices in $G(\bar{A})$ is irrelevant to the properties of the graph itself. Relabelling the vertices has no impact on the digraph intrinsic properties but produces a simultaneous row and column permutation on the corresponding incidency matrix. Such permutations can be used to reorder the associated matrix. A good reordering can lead the matrix to a block triangular form using the concept of strong components of a digraph.

Up to now, we have derived a matrix from a digraph. Our problem, however, requires the inverse procedure *i.e.* to know how to derive a graph from a matrix. For this reason, we need to be acquaint with the concept of structure of a sparse matrix.

Real matrices, *e.g.* Jacobian matrices, are often sparse matrices. The structure of a sparse matrix, also called structure matrix, is the relative position in the matrix of the null and non-null elements. The structure matrix can be represented by a Boolean matrix simply by changing the non-null elements by the number 1 and keeping the null elements as 0.

This Boolean matrix can be considered as the incidency matrix of an associated graph and, with some abuse of terminology, the incidency matrix of the real matrix. Figure 2 shows a simple digraph whose incidency matrix is shown in Fig. 3 on the left. The real matrix, Fig. 3 on the right, is a matrix whose associated digraph is shown in Fig. 2.

Vertices of an associated digraph can be related to the input and output variables of the linear system represented by eq. (1.1) *i.e.* to the velocities at

Cartesian and joint space. This association will be described in more detail in section 2.3. Up to now, it is sufficient to state that diagonal blocks of the hierarchical canonical form of the Jacobian, as obtained in eq. (1.4), have intrinsic relationship to the role that these associated variables play on the sequentially organized inversion of the Jacobian *e.g.* eqs. (1.5)- (1.7).

There is an intrinsic connection between the diagonal blocks and the strong components of the digraph. The relative positions of these diagonal blocks have also a correspondence with the partial order among these diagonal blocks. These concepts will be explained below.

2.2.2 Strong components of a digraph

The main concept used in this work related to directed graphs is the concept of strong components [Harary, Norman e Cartwright 1965]. This concept will be derived here as an extension of the concept of reachability.

To reach the concept of reachability we have to define some preliminary graph concepts. A *path* ia a sequence of arcs $\{e_1, e_2, \ldots\}$ where the initial vertex of the succeeding arc (e_{i+1}) is the final vertex of the preceding arc (e_i). The *length* of a path is the number of arcs in the path. A path comprising a single arc has unit length. Let two vertices i and j be part of a same directed graph G. Vertex j is said *reachable* from vertex i, symbolically $i \rightharpoonup j$, if exists a (directed) path from vertex i to vertex j.

Two vertices i and j are *strongly connected* if exists a path from vertex i to vertex j as well as a path from vertex j to vertex i. Two strongly connected vertices i and j will be designated by the symbol $i \rightleftharpoons j$ like $2 \rightleftharpoons 3$ in Fig. 2. By definition a vertex is always strongly connected to itself $(i \rightleftharpoons i)$. The concept of strongly connected vertices can be extended from a pair of vertices to a generic set of vertices. A subgraph is a *strong component* of the digraph or a *strongly connected set* of vertices if every two vertices are strongly connected.

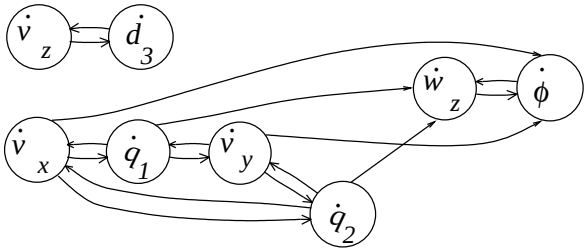

Figure 4: Digraph of the Scara Jacobian according to eq. (1.2)

In a strong component, any pair of vertices, i and j, can be randomly selected and there will be at least two paths connecting these vertices, one for each orientation.

A subset of vertices strongly connected to a given vertex i form a strong component $K(i)$ within the set of all vertices of the graph. The vertex set V of graph G can then be partitioned into strong components $K(i)$ which are, by definition, disjoint.

Eq. (1.2) has the associated digraph shown in Fig. 4 from where we can list the strong components

$$\{v_x, \dot{\theta}_1, v_y, \dot{\theta}_2\}, \{v_z, \dot{d}_3\}, \{\omega_z, \dot{\phi}\}, \tag{2.4}$$

Some strong components in eq. (2.4) show a explicit dependency of other strong components *e.g.* the strong component $\{v_z, \dot{d}_3\}$ is dependent upon the strong component $\{v_x, \dot{\theta}_1, v_y, \dot{\theta}_2\}$. In fact, the strong components of a graph are hierarchically ordered according to a partial order that will be here defined:

Partial order $(\prec)$ is a relationship between two strong components $K(i)$ and $K(j)$.

$K(i) \prec K(j) \iff$ vertex $i \in K(i)$ is reachable for some vertex $j \in K(j)$ and $K(i) \neq K(j)$

A vertex $i \in G$ in a strong component $K(i) \subset G$ is the base (*i.e.* the initial vertex) of a path that can reach any other vertex $j \in K(i)$. To maintain a coherent notation $K(i) = K(j)$ if $i, j \in K(i)$.

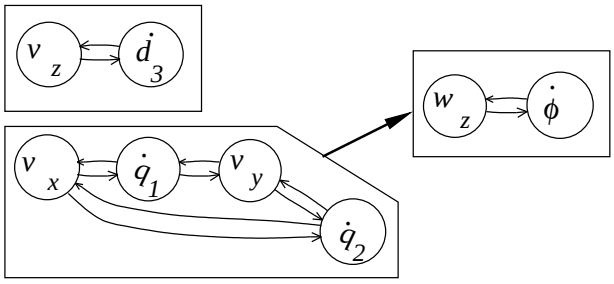

Figure 5: Condensation of the digraph related to the Scara Jacobian

Partial order leads to the concept of *condensation* of a digraph. The condensation of a digraph is a digraph whose vertices are the strong components of the digraph. The condensation of a digraph is more precisely a tree because a loop or cycle in the condensation imply that the strong components were not well determined.

A tree, like Fig. 5, has sources and sinks [Harary, Norman e Cartwright 1965] which corresponds to the beginning and the end of the tree. Sources are the variables which correspond to a vertex in the condensation digraph with no output arcs. Sinks, on the other hand, are variables which correspond to a vertex in the condensation digraph with no input arcs.

The source variables have priority on the solution of eq. (1.1). Variables $\dot{\theta}_1, \dot{\theta}_2; v_y, v_x$ are the sources of the Scara. Joint velocities and end-effector velocities are separated by a semi-colon ";" as they play different roles on the forward and inverse kinematics. Sources must be solved first in the inverse kinematic problem.

The sink variables are those which have complete dependency of the remaining variables. In the inverse kinematic problem they are solved at the final steps of the algorithm. Variables $\dot{\phi}$ and ω_z are the sinks of the Scara.

An isolated vertex in the condensation digraph is at same time a source and a sink. Vertex B in Fig. 5 is isolated and points out that variables v_z and $\dot{d}_3$ are completely independent from the remaining variables. These isolated vertices can be solved at any time because they have no influence on the remaining variables

of the system.

2.2.3 Related matrices $\bar{R}, \bar{S}, \bar{T}$

In the prior sections, we have shown the relationship among the incidency matrix, the associated digraph and the block-triangular form. The block-triangular form is derived from the strong components of the associated digraph. There are several methods to extract the strong components of a digraph and they can be divided into two groups: those directly based on a computational structure which represents the graph itself and those based on the incidency matrix representation of the digraph.

The first group covers methods which rely on linked chains or object-oriented analysis [Orwant, Hietaniemi e Macdonald 1999]. Other methods of the first group are those based on a different graph theoretical basis for the representation of the system, like methods based on bipartite graphs [Murota 1987] [Harary 1967]. Both approaches require some expertise of the programmer as well as the availability of the software tools.

While methods based on the structure of the graph itself have some advantages over the incidency matrix methods they lack some didactic appeal. Moreover their use impose strong prerequisites both at conceptual level and at programming level. Such requisites can puzzle researchers not directly concerned with some subtleties of graph theory as, for instance, roboticists. On the other hand, robot researchers are normally well trained in the use of simulation softwares, specially those matrix orientated, such as Matlab [Corke 1996]. This fact is improved by the existence of free matrix-based simulation softwares such as Octave [Eaton 1996], which is the simulation software used to obtain all results in this paper.

For this reason we take a route that converges to the matrix-based Algorithm 1. This algorithm is based on the work of Reinschke [Reinschke 1988] and it is entirely based on manipulations of the incidency matrix. The remaining of

this section intends to clarify these manipulations and to interpret the intermediary matrices generated by the Algorithm 1.

First, we define the reachability matrix $\bar{R}$ that maps the "inter-accessibility" of vertices. The matrix $\bar{R} = [r_{ij}]$ can be defined element by element, as

$$r_{ij} = \begin{cases} 1 & \text{if a path leads from vertex } j \text{ to vertex } i \\ 0 & \text{if, otherwise, vertex } i \text{ cannot be reached starting from vertex } j \end{cases} \tag{2.5}$$

Harary *et al.*(1965) discuss how to obtain the reachability matrix from the incidency matrix based on some properties of the product of Boolean matrices and power of Boolean matrices in terms of their corresponding associated digraphs. A discussion about the properties of the power of incidency matrices is presented at the Appendix C. This discussion derives the reachability matrix from the incidency matrix using the following summation

$$\bar{R} = \sum_{k=0}^{n-1} \bar{A}^k \tag{2.6}$$

As matrix $\bar{A}$ is Boolean, eq. (2.6) has multiplication replaced by the Boolean **and** ($\wedge$) and sum replaced by the Boolean **or** ($\vee$). The same definitions are valid for other Boolean matrices like $\bar{S}$ and $\bar{T}$ described below.

If the reachability matrix is full then a path starting at vertex i and ending at a different vertex j can *always* be found. The interdependency of all variables is complete. In matricial terms this means that there is no simultaneous permutation of rows and columns that lead this matrix to a block-triangular form.

A full reachability matrix is a signal of stop. Nothing more can be done to reduce the initial incidency matrix to an upper block-triangular form. A full incidency matrix, by eq. (2.6), implies in a full reachability matrix. However the converse is not true: a full reachability matrix can be obtained from a sparse incidency matrix *e.g.* the digraph in Fig. 2 is fully reachable and the reachability matrix derived from the sparse matrix $\bar{A}$ in Fig. 3 is full.

When the reachability matrix has some zeros a block-triangular form can *always* be reached [Harary, Norman e Cartwright 1965]. In this case, two other Boolean matrices, $\bar{S}$ and $\bar{T}$, help us to reduce the original matrix $\bar{A}$ to an upper block-triangular form.

The *symmetrical structure Boolean matrix* $\bar{S}$ shows the strong component $K(i)$ associated with each vertex i. Matrix $\bar{S}$ comes directly from the reachability matrix $\bar{R}$ by

$$\bar{S} \triangleq \bar{R} \wedge \bar{R}^T \tag{2.7}$$

i.e. each element s_{ij} from $\bar{S}$ satisfy

$$s_{ij} = r_{ij} \cdot r_{ji} \tag{2.8}$$

The matrix $\bar{S}$ maps all strong components of vertices in a directed graph. A column j of $\bar{S}$ (which corresponds to vertex j) has "connections" with all vertices i whose elements $s_{ij} \neq 0$, $i = 1, n$. Strong components $K(j)$ are formed by all vertices i such that $s_{ij} \neq 0$. By definition, two strong components with the same number of variables are either disjoint or congruent *i.e.* are the same strong component.

The symmetrical structure matrix $\bar{S} = [s_{ij}]$ can be alternatively defined by

$$s_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are strongly connected.} \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

The *asymmetrical structure Boolean matrix* $\bar{T}$ differentiates strong components one another.

$$\bar{T} \triangleq \bar{R} \wedge \backsim \bar{R}^T \tag{2.10}$$

where the operator $\backsim$ is the Boolean *not* applied to all elements of the Boolean matrix. The matrix $\backsim \bar{R}$ is simply a change $0 \rightleftharpoons 1$ of all elements of the original Boolean matrix $\bar{R}$.

In other words

$$t_{ij} = r_{ij} \cdot \backsim r_{ji} \tag{2.11}$$

The matrix $\bar{T} = [t_{ij}]$ can be similarly defined as

$$t_{ij} = \begin{cases} 1 & \text{if a path leads from vertex } j \text{ to vertex } i \\ & \text{but there is no return path } i \rightarrow j \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

This interpretation of matrix $\bar{T}$ leads to the definition of the asymmetry vector $\vec{w} = [w_1, w_2, \cdots, w_n]$ where

$$w_j \triangleq \sum_{i=1}^{n} t_{ij} \tag{2.13}$$

The summation in eq. (2.13) is *not* Boolean. So w_j is the number of vertices k that can be reached *starting from* vertex j but no return path from k to j exists. The vector $\{w_j\}$ is formed by the sum of all elements in column j of matrix $\bar{T}$.

If two vertices i and j are on the same strong component then $w_i = w_j$; however, the reciprocal condition is not true. If $w_i = w_j$, *i.e.* if both vertices i and j have the same degree of asymmetry, it is *not* guaranteed that they are on the same strong component. Sets of vertices with the same degree of asymmetry, $w_i = \text{const}$, can be possibly split up into smaller strong components using the symmetrical structure matrix $\bar{S}$.

The elements of $\vec{w}$ must be rearranged according to their magnitude (in crescent order) keeping record of the original indices. The vertices with the same value of w_j are arranged in crescent order. Inside these groups all vertices of a same strong component must be kept in sequence.

The rearranged vector $\tilde{w} = P\vec{w}$ is a permutation of the original vector $\vec{w}$. The same permutation matrix P converts the incidency matrix $\bar{A}$ to $\check{A} = P\bar{A}P^T$ where $\check{A}$ is block triangular [Reinschke 1988].

Algorithm 1 describes how to obtain the block-triangular form of the kinematic structure matrix. Notice that the row permutation (P) applied to $\bar{A}$ is the same column permutation (P^T) in itens 6c and 8 of Algorithm 1.

Algorithm 1: Graph partitioning based on Boolean matrices Reinschke (1988)

- Input data: A
- Output data: reordered matrix $\check{A}$

1. Incidency matrix: $\bar{A}$ from original matrix A
2. Reachability matrix: $\bar{R} = \sum_{i-1}^{n-1} \bar{A}^k$
   - If $\bar{R}$ is full, stop. No improvement can be obtained by permutating rows or columns.
   - else, continue.
3. Symmetrical structure matrix: $S \triangleq R \wedge R^T$
4. Asymmetrical structure matrix: $T \triangleq R \wedge \bar{R}^T$
5. Asymmetry vector: $\vec{w} : w_j = \sum_{i=1}^{n} t_{ij}$
6. Reordering of the asymmetry vector $\vec{w}$
   (a) Reorder the vertices with the same value of w_j in crescent order.
   (b) Check inside each group of vertices with the same value of w_j if the group corresponds to more than one strong component. This check-up is done by looking at the corresponding columns of matrix S.
      - If these columns are absolutely equal go to item 7.
      - Else divide these groups: one for each pattern of the columns of S.
   (c) Reorder these groups internally keeping all vertices of a same strong component in sequence.
7. Find the permutation matrix P that leads $\{w_1, w_2, \cdots, w_n\} \rightarrow \{w_{e_{ind_1}}, w_{e_{ind_2}}, \cdots, w_{e_{ind_n}}\}$ where $\{e_{ind_1}, e_{ind_2}, \cdots, e_{ind_n}\}$ is the sequence of permuted indices of the original vector.

$$P = \begin{bmatrix} \cdots & \vec{e}_{ind_1} & \cdots \\ \cdots & \vec{e}_{ind_2} & \cdots \\ & \vdots & \\ \cdots & \vec{e}_{ind_n} & \cdots \end{bmatrix} \tag{2.14}$$

   where $\vec{e}_{ind_j}$ is a row vector corresponding to the j-th row of the identity matrix I_n.
8. Reordering of the original matrix A : $\check{A} = PAP^T$

2.3 Hierarchical Jacobian reordering

Simultaneous rows and columns permutations are too restrictive to obtain the block-triangular form of a generally sparse Jacobian matrix. For this reason, digraph techniques, when directly applied to the Jacobian matrix, have poor results.

To overcome this deficiency, we define the kinematic structure matrix Q as

$$Q \triangleq \begin{pmatrix} 0 & J \\ J^{-1} & 0 \end{pmatrix} \Longrightarrow \begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} = Q \begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} \tag{2.15}$$

The matrix Q is part of a linear system with identical input and output vectors. The i-th column and i-th row are associated to the same variable in $\left[\dot{x}^T \ \dot{q}^T\right]^T$. This property turns the restrictive simultaneous rows and columns permutations on Q be as general as independent rows and columns permutations on J.

The matrix Q can be reordered following the procedures described in section 2.2.2. These procedures basically rely on Boolean matrices products and sums and can be performed in any numerical software.

The reordering of Q, eq. (2.15), is more properly a simultaneous reordering of both matrices J and J^{-1}. The Puma robot in section 4.1 illustrates the use of these concepts. The matrix Q need not to be *a priori* known but only its structure matrix

$$\bar{Q} \triangleq \begin{pmatrix} 0 & \bar{J} \\ \overline{J^{-1}} & 0 \end{pmatrix} \tag{2.16}$$

where $\bar{J}$ and $\overline{J^{-1}}$ are incidency matrices of the Jacobian and inverse Jacobian matrices, respectively.

While eq. (2.16) requires the prior knowledge of the incidency matrix of the inverse Jacobian, $\overline{J^{-1}}$, the exact values of J^{-1} do not need to be known; therefore, the incidency matrix $\overline{J^{-1}}$ can be obtained without the symbolic inversion of J. Several methods exist to obtain $\overline{J^{-1}}$ either using the graphical interpretation of the determinant [Jantzen 1996] or, numerically, by inverting a series of matrices

whose incidency matrix is $\bar{J}$ as described in [Martins 2000].

Let $\check{Q}$ be the hierarchical canonical form of the kinematic matrix Q, which is obtained by applying Algorithm 1. Once $\check{Q}$ is obtained we have at the same time the hierarchical canonical form of the Jacobian and also of the inverse Jacobian. For instance, to obtain the hierarchical canonical form of the Jacobian matrix is sufficient to drop the lines (rows and columns) which corresponds to the inverse Jacobian, see the example in section 4.2.

3 *Jacobian Matrices and Screws*

There are two main approaches for obtaining the Jacobian [Tsai 1999]: Denavit-Hartenberg method and screw-based method. Section 3.2 describes the Denavit-Hartenberg method. The screw-based method will be described from section 3.4 on.

Section 3.3 is intermediary between both descriptions, and shows how the velocities of a link in a serial robot depend upon the joint variables. This description is used in section 3.4 to present the concepts of screw theory. Section 3.5 describes how the screws depend upon the joint variables, expanding the results of section 3.3.

3.1 Sparsity of the Jacobian matrix

The rearrangement of matrices to their block-triangular form using graph techniques entail sparse matrices [Pissanetsky 1984]. An extreme case is when the associated digraph of the matrix is strongly connected, section 2.2.2, so any trial to find a block-triangular form from such matrices is worthless. The opposite extreme case is when the matrix is upper triangular. In this case, the matrix is already in the hierarchical canonical form; any permutation of this matrix will not improve its form.

The Jacobian matrix is not unique for a robot architecture; on the contrary, it is highly dependent upon the coordinate system. To be profitable, the kinematic

analysis must explore and enhance the sparsity of the Jacobian matrix as much as possible *before* using graph theory.

3.2 Denavit-Hartenberg method

Classical texts in Robotics [Asada e Slotine 1986] [Spong e Vidyasagar 1989] present the Denavit-Hartenberg method as *the* method of robot kinematic analysis. Later, other texts [Angeles 1997] [Tsai 1999] present the Denavit-Hartenberg method in parallel with screw-based analysis method. Although simple and easy to apply to most of the serial robots, the Denavit-Hartenberg method has some important drawbacks, *e.g.* the Jacobian is less sparse and has elements with higher complexity [Hunt 1987] [Zomaya, Smith e Olariu 1999].

The essence of Denavit-Hartenberg method is the systematic generation of a series of coordinate systems for the robot. These coordinate systems are based on geometrical properties of the manipulator as well as the type of the joints of the manipulator.

Due to its algorithmic features, the Denavit-Hartenberg method permits a "blind use" of the kinematic principles and alleviates some problems related to the search of the "best" coordinate system for each manipulator architecture. It is a handy option for a first contact with a different robot architecture, or when inverse kinematics efficiency is not a crucial problem. The Denavit-Hartenberg method also permits a prompt off-line computation of the manipulator model.

The Denavit-Hartenberg method, see *e.g.* [Spong e Vidyasagar 1989], is essentially an algorithm based on the Denavit-Hartenberg parameters [Denavit e Hartenberg 1955]. This algorithm obtains, in a systematic way, each one of the columns of the Jacobian matrix through the multiplication of homogeneous matrices.

These homogeneous matrices relate the movement between the coordinate systems. Apart from the existence of multiple coordinate systems, the Jacobian

matrix is always determined in relation to the first ("0") coordinate system at the base. For this reason, the terms which compose the Jacobian matrix are normally extremely complex.

To minimise the complexity of the Jacobian elements, some authors [Tsai 1999] suggest the use of the "modified" Denavit-Hartenberg method, *i.e.* the use of a mobile reference frame to represent the Jacobian matrix. Waldron *et al.*(1985) suggests the use of an intermediate link to attach the coordinate system. Although with better form than the original Denavit-Hartenberg Jacobian matrix, this "modified" Jacobian matrix is still based on the same fixed set of coordinate systems.

Denavit-Hartenberg coordinate systems are judiciously selected to exploit the geometrical relations between *consecutive links*. However, geometrical relations between *non consecutive links*, which are not considered by Denavit-Hartenberg method, can also improve the sparsity of the Jacobian matrix. Screw-based methods normally take in account both types of relations, so yielding sparser Jacobian matrices.

In the Denavit-Hartenberg method, most of the coordinate systems are uniquely determined. This rigid selection of the coordinate systems affects directly the sparsity of the obtained Jacobian matrix [Hunt 1987]. It is known from the literature that well chosen coordinate systems improve the sparsity of the Jacobian [Wang e Waldron 1987] [Waldron, Wang e Bolin 1985]. Screw theory concepts lead to a more suitable choice of the coordinate system [Hunt 1987]. For these reason, we will adopt a screw theory approach to derive the Jacobian matrix throughout this work.

3.3 Velocities along a serial chain

This section describes the kinematic behaviour of a serial chain. A serial (kinematic) chain in this context is a series of links connected by joints with one degree of freedom. Each link in the chain is labelled by a natural number where

0 is the base (fixed frame) and n is the end-effector. The objective of this section is to show how the linear and angular velocities at a generic link i is influenced by the linear and angular velocities of the previous joints.

Although the example used here is a serial manipulator, the results are quite general and will be applied to redundant robots, chapter 5, and with some adaptation to parallel robots, chapter 6.

Let a serial manipulator in the imminence of being instantaneously displaced. First, all kinematic pairs in the serial chain are fixed. The whole manipulator becomes kinematically equivalent to a single rigid body, Fig. 6, satisfying the rigidity principle discussed in appendix A.1.

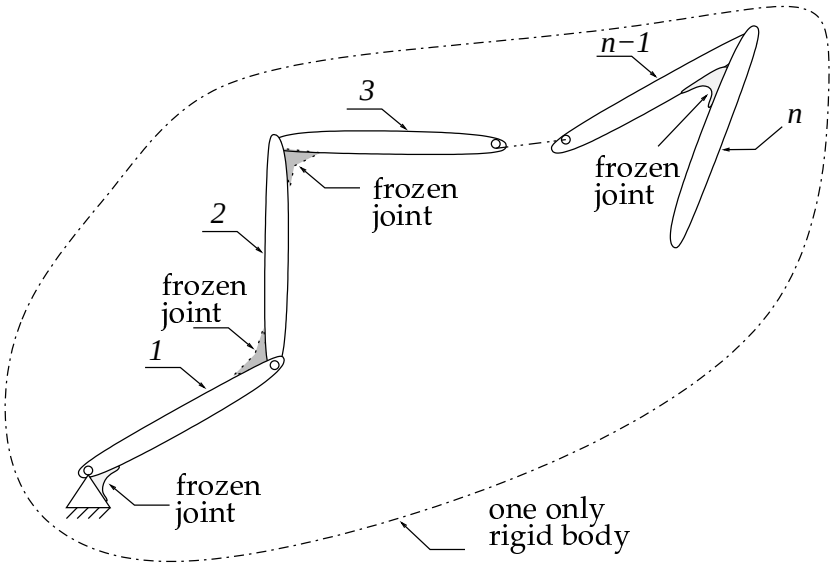

Figure 6: A manipulator with all joints frozen behaving as a rigid body B_{0-n}.

All joints, now instantaneously "frozen", are considered helical kinematic pairs. A helical kinematic pair can be visualised as a screw attached to one link displacing relative to a nut attached to the other link. This screw nut pair has a pitch h. This pitch can be zero (rotative pair), infinite (prismatic pair) or any other finite value. In other words, the helical pair is a generalisation of all kinematic pairs with one degree of freedom that have contact between two mating surfaces (lower kinematic pairs, see [Hartenberg e Denavit 1964]).

The joint n between link $n-1$, and the end-effector, link n, is now released, Fig. 7. An actuator twists this joint with magnitude $\dot{q}_n$. The "rigid body" $B_{0-(n-1)}$, a virtual link composed by links 0 to $n-1$, remains fixed and, as expected, is not affected by this input variable $\dot{q}_n$. The remaining of the manipulator, namely link n, is still a single rigid body being twisted about the screw kinematic pair of the joint n, *i.e.* link n translates and rotates by the sole imposition of joint n.

If joint $n-1$, instead of joint n, is the only joint released a similar result occurs, Fig. 8. Joint $n-1$ connects links $n-2$ and $n-1$. Two rigid bodies are divided by joint $n-1$, namely $B_{0-(n-2)}$ and $B_{(n-1)-n}$. Body $B_{0-(n-2)}$, composed by links 0 to $n-2$, is fixed. Body $B_{(n-1)-n}$, composed by links $n-1$ and n, displaces relatively to body $B_{0-(n-2)}$ *i.e.* relatively to the base.

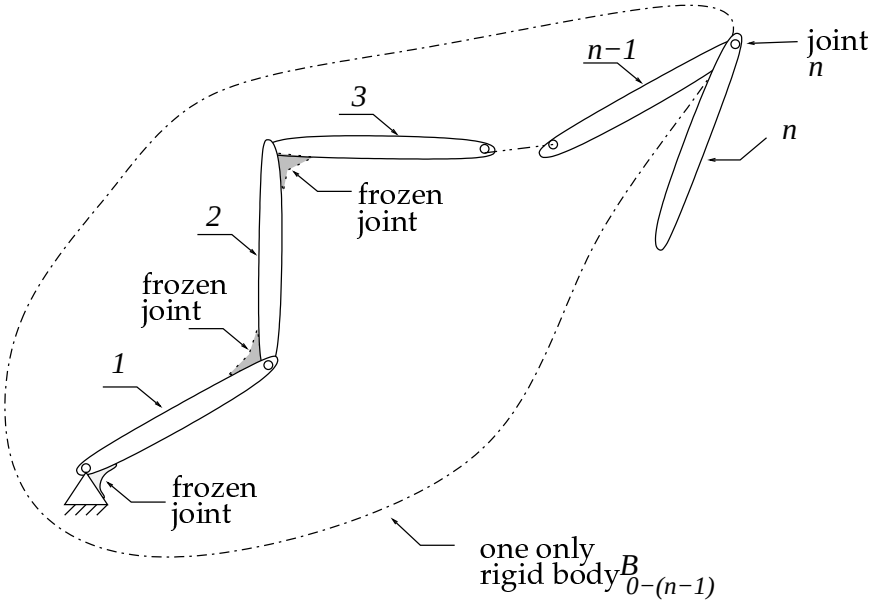

Figure 7: A manipulator with all joints frozen with the exception of joint n between the last and penultimate links. There are two rigid bodies: the end-effector n, and $B_{0-(n-1)}$ composed by the remaining frozen bodies.

If the same reasoning is applied to all joints some conclusions can be extracted about the translational and angular velocities[1] of a link i:

[1] A discussion of translational and angular velocities in the context of screw theory can be

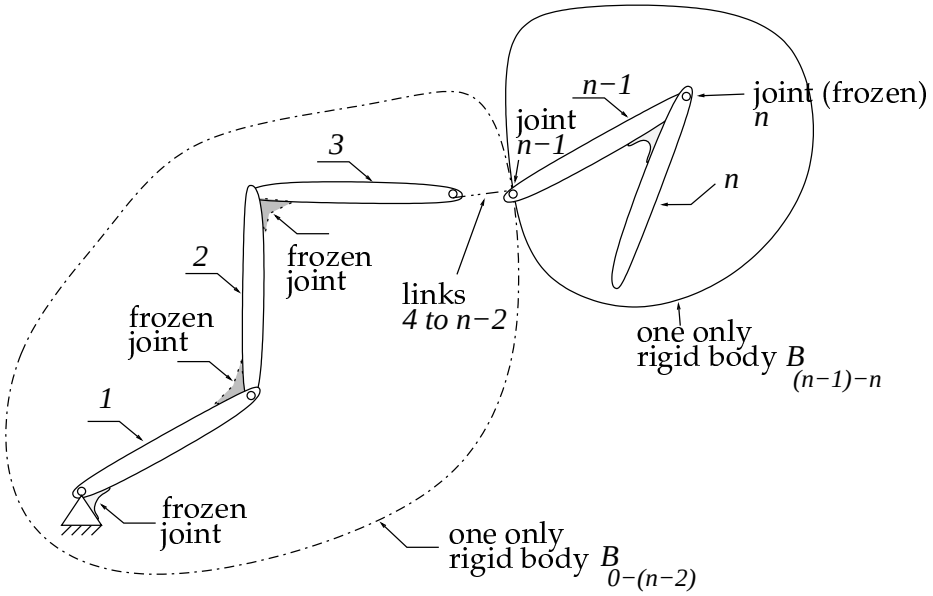

Figure 8: A manipulator with all joints frozen with the exception of joint $n-1$ between links $n-2$ and $n-1$. There are two rigid bodies: $B_{(n-1)-n}$ composed by links $n-1$ and the end-effector n and the remaining frozen bodies $B_{0-(n-2)}$

- The translational and angular velocities of link i is affected by the twists imposed on joints 1 to i, *i.e.* those joints which are between the link i and the base 0. The remaining joints do not affect velocities of the link i.
- The angular velocity of body i is the *vectorial sum* of the angular velocities applied at joints 1 to i.

Let P be a fixed point on the link i. As the velocity of a point is a vector we also conclude that:

- The linear velocity of a point P at link i is the *vectorial sum* of the linear velocities imposed by each of these joints, as if they were acting alone and independently.

A convenient choice for the point P is the origin of the coordinate system, this point becomes unambiguously defined whatever link i is chosen. The rigid

found in appendix A and in [Ciblak 1998].

body movement of a link is completely determined in a given coordinate system when

- The angular velocity ω of the link is represented in this coordinate system, and
- The linear velocity v_O of a point which is instantaneously coincident with the origin O of this coordinate system is also known.

3.4 Screw-based Jacobian matrix

This section shows how to generate the Jacobian matrix using the screw theory principles. This generation is based on two fundamentals: the linear properties, and the separation of the magnitude of the screws from their geometric information. The former was described in the last section, while the latter will described in the sequence.

Screw theory is based on the physical principles of the rigid kinematics. Rigid body instantaneous movements can be "carried out by" a screw [Ball 1900]. A screw is a geometrical element with an axis (line), a pitch and a magnitude in velocity dimensions (rad/s or m/s). From the conclusion of the last section, the rigid body instantaneous movement of a link can be described by a pair of vectors ω and v_O, so a screw $\$_i$ representing the movement of link i can be represented by the pair $\$_i = (\omega_i; v_O|_i)$. Using screw theory principles [Hunt 1978] [Tsai 1999], we can extract the geometrical parameters of the screw (axis, pitch and magnitude) from the representation $\$_i = (\omega; v_O|_i)$.

Although the helical pair is the generalisation, in common practice only rotative and prismatic pairs are used. For these cases, Table 1 summarises the values of axis, pitch and magnitude. There $\hat{\$}$ is the normalised screw associated with the joint, $\hat{s}$ is a unit vector in the direction of the axis, and s_o is a vector starting at the origin of the coordinate system and ending in any point along the screw axis.

joint i	q_i	$\hat{\$}$	h
rotative	ω	$\begin{pmatrix} \hat{s} \\ s_o \times \hat{s} \end{pmatrix}$	0
prismatic	v	$\begin{pmatrix} 0 \\ \hat{s} \end{pmatrix}$	∞

Table 1: Normalised screws $\hat{\$}$, pitches h and magnitudes q_i for rotative and prismatic pairs

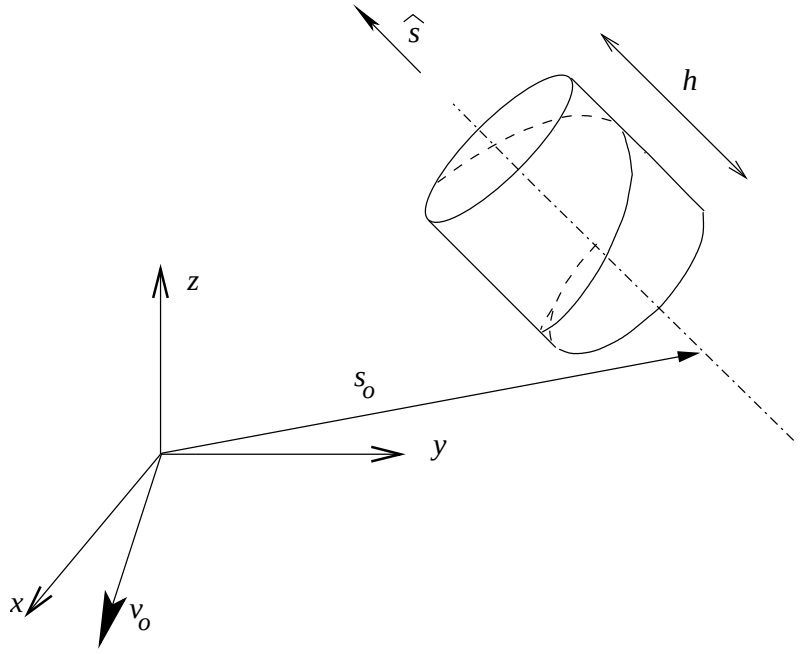

Figure 9: A normalised screw $\hat{\$}$ represented by vectors $\hat{s}$ and s_o in a generic coordinate system. The pitch h and the velocity at the origin v_o are also shown.

The separation of a screw as a product between the normalised screw and the magnitude, $\$_i = \hat{\$}_i \dot{q}_i$ allows the determination of the Jacobian matrix almost directly, sometimes by simple inspection of the robot geometry [Hunt 2000]. The instantaneous movement of the end-effector $\$_n$ is the sum of the instantaneous movements, $\$_i$, of the links composing the serial chain.

$$\dot{x} = \$_n = \sum_{i=1}^{n} \$_i = \sum_{i=1}^{n} \begin{bmatrix} J_{R_i} \\ J_{L_i} \end{bmatrix} \dot{q}_i = \sum_{i=1}^{n} \hat{\$}_i \dot{q}_i \tag{3.1}$$

where $\dot{x} = \$_n = [\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6]^T$ is represented in an appropriate coordinate system, $J_{R_i} = [\dot{x}_1, \dot{x}_2, \dot{x}_3]^T|_i$, $J_{L_i} = [\dot{x}_4, \dot{x}_5, \dot{x}_6]^T|_i$. Each column of the Jacobian is a *normalised* screw $\hat{\$}_i$ of eq. (3.1) corresponding to the i-th joint. More details about this topic can be found in [Hunt 1987] and references therein.

Isolating vector $\dot{q}$ in eq. (3.1) we obtain the system

$$\dot{x} = \$_n = J\dot{q} \tag{3.2}$$

with the Jacobian matrix defined by

$$J = [\hat{\$}_1 \ \hat{\$}_2 \ \hat{\$}_3 \cdots \hat{\$}_n] \tag{3.3}$$

Two details about eq. (3.1): vectors J_{R_i} and J_{L_i} are consequence of the linear properties described in section 3.3, and in the Denavit-Hartenberg method normally the order of the normalised screws is $(v; \omega)$ changing the order of the vectors J_{R_i} and J_{L_i}.

Screw theory is more flexible regarding the choice of the coordinate system than the Denavit-Hartenberg method. In fact, the coordinate system for representing the screws can be freely chosen and then all joint velocities are represented in relation to this coordinate system. Furthermore, we can migrate from one coordinate system to another just using one matrix multiplication *i.e.* the screw transformation matrix [Hunt 1987]. The screw transformation matrix, here in ray order and from coordinate system i to coordinate system j, is [Tischler et al.

2000]

$$\mathbf{T}_{i\to j} \triangleq \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{RS} & \mathbf{R} \end{bmatrix} .\text{where } \mathbf{S} \triangleq \begin{bmatrix} 0 & -x_o & y_o \\ x_o & 0 & -z_o \\ -y_o & z_o & 0 \end{bmatrix} \tag{3.4}$$

The matrix $\mathbf{T}_{i\to j}$ transforms the Jacobian matrix, originally given in a coordinate system i, into a different coordinate system j *i.e.* $J_j = \mathbf{T}_{i\to j} J_i$. In eq. (3.4) $\mathbf{R} = R_{i\to j}$ is the 3×3 rotation matrix from coordinate system i to coordinate system j [Spong e Vidyasagar 1989], and (x_o, y_o, z_o) are the coordinates of the origin of the coordinate system j measured in the coordinate system i.

It must be recalled that, the Jacobian matrix, while different in form from one coordinate system to another, has the same functionality. The screw transformation matrix $\mathbf{T}_{i\to j}$, eq. (3.4), couples, or decouples, the terms that were not present in other Jacobian representations,*e.g.* the Denavit-Hartenberg Jacobian matrix. The main difference is that matrix $\mathbf{T}_{i\to j}$ is orthogonal, and then is easily invertible

$$\mathbf{T}_{i\to j}^{-1} = \begin{bmatrix} \mathbf{R}^T & (\mathbf{RS})^T \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix}. \tag{3.5}$$

where $(\mathbf{RS})^T = \mathbf{S}^T\mathbf{R}^T = -\mathbf{S}^T\mathbf{R}$.

To select a good coordinate system, *i.e.* a coordinate system which improves the sparsity of the Jacobian, some geometrical insight is required. Once this coordinate system is chosen, we apply graph theory techniques to prepare the Jacobian matrix to the inversion process. The graphical techniques were described above

To guide the selection of an appropriate coordinate system, the next section will describe how the screw in a serial chain depend upon of the joint variables.

3.5 Screw dependence of the joint variables

Screws are instantaneous "shots" of the movement of a rigid body, and they are described in a specific coordinate system. There are several ways to represent

the (instantaneous) movement of a rigid body, such as quaternions, homogeneous matrices and dual numbers [Katz 1997], but all of these methods also need a coordinate system to represent their elements. So a judicious selection of the coordinate system is a important part of the kinematic analysis of a manipulator.

Denavit-Hartenberg methods, which are based on homogeneous matrices, set the coordinate system fixed on the base. Screw theory, on the other hand, represents the screws in a coordinate system fixed in any of the links of the chain, in general one of the mobile links. Using the screw transformation matrix, eq. (3.4), the screw can be easily represented in any other coordinate system, *e.g.* at the base or at the end-effector.

The relative movement of two consecutive bodies in a kinematic chain is determined by the joint between them, and the screw axes coincide with the joint axes. In a serial manipulator, each link is binary *i.e.* has two joints attached. When the coordinate system is attached to a link in a serial manipulator, these two screws (joints) axes remain fixed relatively to the coordinate system. Other screws (joints) axes, however, change their relative positions when some of the joint angles change. This fact will be explored in the sequence.

The columns of Jacobian matrix are screws, eq. (3.3), so the Jacobian matrix depends upon the coordinate systems which represent the screws for each joint of the serial manipulator. Waldron *et al.*(1985) and Wang & Waldron (1987) describe how the columns of a Jacobian matrix are dependent upon certain variables (*e.g.* joint angles) and do not depend upon others.

Figure 10 illustrates the dependence of the screws according to the coordinate system. If the coordinate system is at link i then the screws associated to the joints i and $i+1$ became fixed. Other links have movement relative to link i, so the screws fixed at these joints depend upon the joint variables connecting these links to the link i. Fig. 11.

For instance when the reference frame is at the base frame (frame 0 following

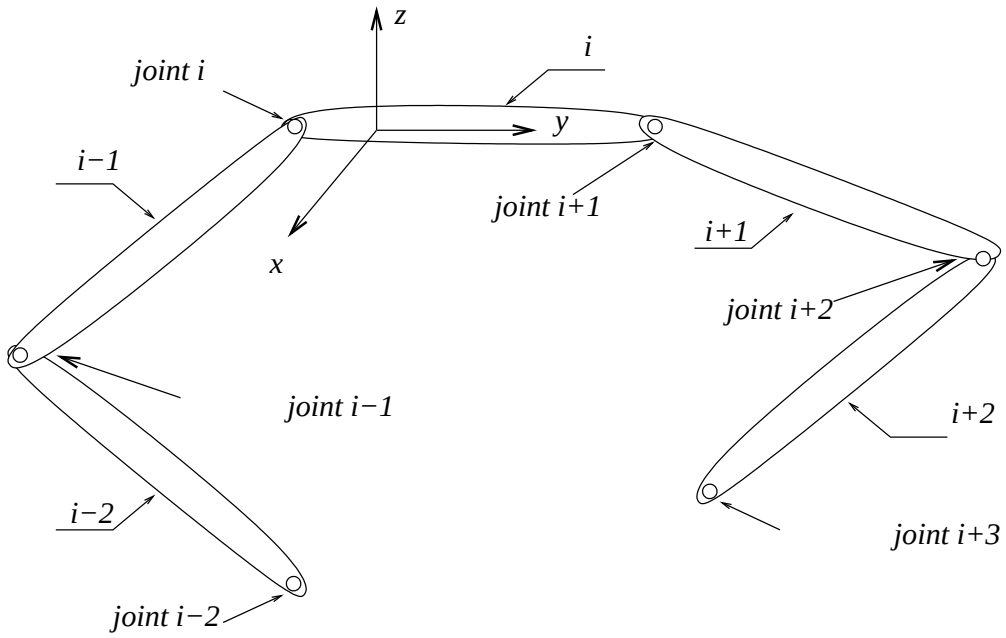

Figure 10: Sequence of links in a serial kinematic chain with the coordinate system fixed at link i.

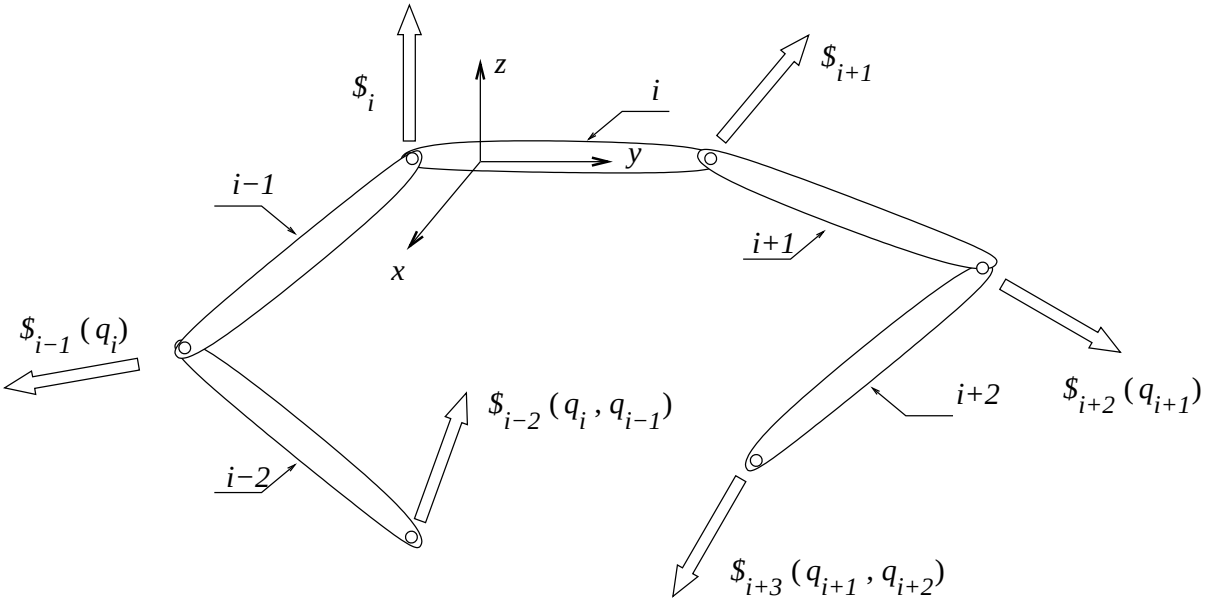

Figure 11: Sequence of links in a serial kinematic chain with the coordinate system fixed at link i making explicit the screw dependence of the joint variables $\dot{q}_i$. Mobile links are filled in black.

Denavit-Hartenberg rules) the dependence table is

$$\begin{aligned}
{}^0\$_1 &= {}^0\$_1 \\
{}^0\$_2 &= {}^0\$_2(q_1) \\
{}^0\$_3 &= {}^0\$_3(q_1, q_2) \\
{}^0\$_4 &= {}^0\$_4(q_1, q_2, q_3) \\
{}^0\$_5 &= {}^0\$_5(q_1, q_2, q_3, q_4) \\
{}^0\$_6 &= {}^0\$_6(q_1, q_2, q_3, q_4, q_5)
\end{aligned} \tag{3.6}$$

where ${}^k\$_j(q_i)$ means that ${}^k\$_j$ is a function of q_i, displacement (angular or translational) at joint i, with a coordinate system attached to link k. If the reference frame is set at the end-effector (frame n following Denavit-Hartenberg rules) then

$$\begin{aligned}
{}^n\$_6 &= {}^n\$_6 \\
{}^n\$_5 &= {}^n\$_5(q_6) \\
{}^n\$_4 &= {}^n\$_4(q_6, q_5) \\
{}^n\$_3 &= {}^n\$_3(q_6, q_5, q_4) \\
{}^n\$_2 &= {}^n\$_5(q_6, q_5, q_4, q_3) \\
{}^n\$_1 &= {}^n\$_6(q_6, q_5, q_4, q_3, q_2)
\end{aligned} \tag{3.7}$$

In either frame, eq. (3.6) or eq. (3.7), the Jacobian matrix is dependent upon five variables $(q_2, q_3, q_4, q_5, q_6)$.

If the reference frame is at an intermediary link then the result will be different. Let, for instance, the frame be at link 3 in the Denavit-Hartenberg conven-

tion.

$$\begin{aligned} {}^3\$_1 &= {}^3\$_1(q_3, q_2) \\ {}^3\$_2 &= {}^3\$_2(q_3) \\ {}^3\$_3 &= {}^3\$_3 \\ {}^3\$_4 &= {}^3\$_4 \\ {}^3\$_5 &= {}^3\$_5(q_4) \\ {}^3\$_6 &= {}^3\$_6(q_4, q_5) \end{aligned} \tag{3.8}$$

The Jacobian matrix, based on eq. (3.8), is dependent upon only four variables: (q_2, q_3, q_4, q_5). This can be explained by the screw coordinate transformation matrix (see section A.3) that aggregates some complexity to the exterior aspect of the Jacobian matrix. It cannot however affect its intrinsic properties like manipulator special configurations. A Jacobian matrix cannot lose nor gain any special configuration via a screw coordinates change.

The rank of a matrix cannot be altered when this matrix is multiplied by a full rank square matrix. The transformation matrix $\mathbf{T}_{i\to j}$, eq. (3.4), is *always* invertible. So $\mathbf{T}_{i\to j}$ cannot affect the number or existence of special configurations. Special configurations are uniquely dependent upon the values of geometrical parameters and the displacement at the joints q_i. Special configurations are structural properties of the manipulator. They are unique even when expressed in different ways or in different coordinate systems.

With reference frame at link 4 the above equations take the following form

$$\begin{aligned} {}^4\$_1 &= {}^4\$_1(q_4, q_3, q_2) \\ {}^4\$_2 &= {}^4\$_2(q_4, q_3) \\ {}^4\$_3 &= {}^4\$_3(q_4) \\ {}^4\$_4 &= {}^4\$_4 \\ {}^4\$_5 &= {}^4\$_5 \\ {}^4\$_6 &= {}^4\$_6(q_5) \end{aligned} \tag{3.9}$$

The Jacobian matrix is described by only four variables. Equations like (3.6)-(3.9) express qualitatively the screw dependency of the columns of the Jacobian matrix. Eqs. (3.8) and (3.9) show that the initial and final actuator variables have no effect on the rank of the Jacobian matrix. Eqs. (3.6) and (3.7) have built-in an "extra" screw transformation (to the base or the end-effector, respectively) that is always invertible (see eq. (3.4) and comments in appendix A.3).

A specific robot architectures is associated with certain screw subsystems [Hunt 2000], *e.g.* the spherical wrist. The knowledge of these subsystems can improve the choice of a good coordinate system which may lead the Jacobian matrix to a sparser form. For example, we can use a coordinate system in which the screws take their simpler form, like those listed in [Gibson e Hunt 1990].

3.6 Line permutations of the Jacobian matrix

The Jacobian is originated from the system $\dot{x} = J\dot{q}$, eq. (3.2). This system is preserved after permutations of lines (rows and columns) if vectors $\dot{x}$ and $\dot{q}$ are correspondingly permuted. Line permutations of the Jacobian matrix are divided into rows permutations and columns permutations.

Rows permutations change the order of the equations (3.2), which leads to a correspondent permutation of the vector $\dot{x}$. The columns of the Jacobian matrix, the screws $\hat{\$}_i$, need similar permutations. These screws are represented in a order different from the original ray order, see [Hunt 1978] and appendix A.2.5.

Lucas *et al.*(2000) for instance suggest the change of the first three rows by the last three rows in their Jacobian matrix. Such a change represented, in that case, simply the substitution of ray order by axis order. As commented out in the appendix A.2.5 there is no obligation in the use ray or axis order.

Columns permutations change the summation order in eq. (3.1). Vector $\dot{q}$ abandons its "natural" order : $(\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6)$. It is natural to keep the crescent order of the indices previously defined, but nothing is assured that this indices order was well chosen.

This section justifies the hierarchical kinematic analysis, chapter 2 , which is based on combinatorial methods [Schendel 1989] [Brualdi e Ryser 1991] applied to the Jacobian matrix. It also supports an important aspect of the robot kinematics: the intrinsic geometrical properties of the screws are *not* affected by any real location of rows and/or columns of the Jacobian matrix.

4 *Hierarchical Kinematic Analysis of Serial Robots*

This chapter applies the hierarchical kinematic analysis to a widely spread industrial robot architecture. Puma robots are also one of the most studied architectures found in research papers on robotics. To obtain the hierarchical canonical form of the Jacobian of this robot, the graph- and matrix-orientated techniques described as hierarchical analysis in the prior sections were used.

Section 4.1 describes the Puma robot and the coordinate system to represented its Jacobian matrix. Section 4.2 presents the hierarchical canonical form of the Puma robot.

Although most of the properties of the hierarchical analysis were presented in chapters 1 and 2, the application of this method to a six-degrees-of-freedom industrial robot reveals other aspects of the method. These aspects are discussed in sections 4.3- 4.5.

4.1 Hierarchical analysis of a Puma robot

The Puma robot, Fig. 12, is a serial manipulator with six degrees of freedom. All joints are rotative kinematic pairs. The last three joint axes intersect one another in a single point forming a so-called spherical wrist. Position angles θ_i at the rotative joints are shown in Figure 12. Screws $\$_i$ are aligned with the joint axes and drawn as conical arrows. The chosen coordinate system is represented by the tryad x, y, z, and is considered fixed on link 3.

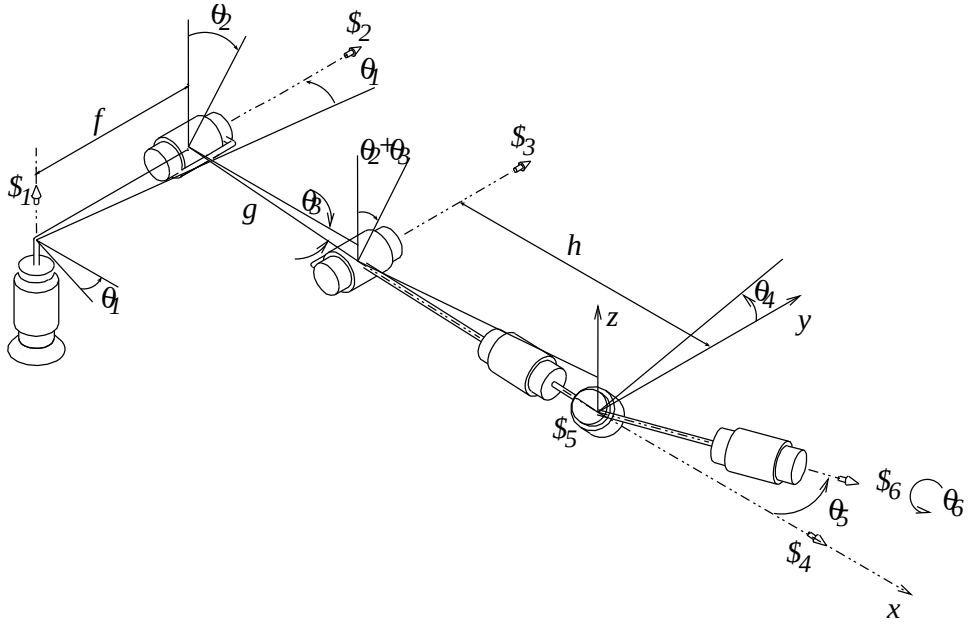

Figure 12: The Puma robot analyzed in this work. The screw $\$_5$, which is not explicitly designated on the Figure is on the joint axis 5 in the middle of the spherical wrist.

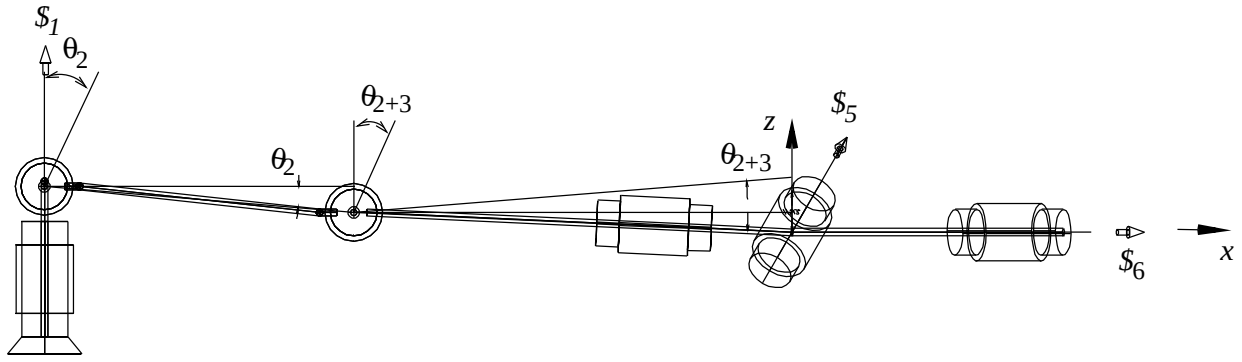

Figure 13: Side view (xz-plane) of the Puma.

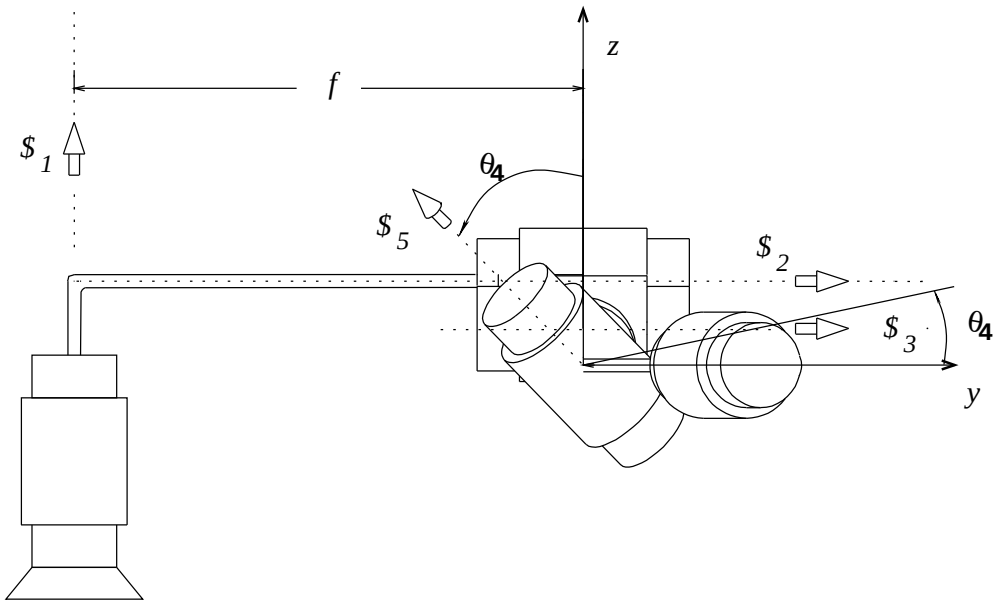

Figure 14: Frontal view (yz-plane) of the Puma highlighting the spherical wrist.

The Jacobian of a Puma robot is a rather complex matrix-valued function, especially when obtained from the Denavit-Hartenberg method. The complexity of the Puma Jacobian imposes selecting some auxiliary variables to obtain their elements recursively [Jr, Nugent e Saridis 1987]. However, using screw theory with a good choice of the coordinate system a significantly sparser Jacobian can be obtained [Hunt 1987]. This Jacobian, with the coordinate system sketched in Figs. 12 and 15, is

$$J = \begin{bmatrix} -s_{23} & 0 & 0 & 1 & 0 & c_5 \\ 0 & 1 & 1 & 0 & -s_4 & c_4 s_5 \\ c_{23} & 0 & 0 & 0 & c_4 & s_4 s_5 \\ -f c_{23} & g s_3 & 0 & 0 & 0 & 0 \\ x_{14} & 0 & 0 & 0 & 0 & 0 \\ -f s_{23} & x'_{41} & -h & 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

where $s_i = \sin\theta_i$; $s_{ij} = \sin(\theta_i + \theta_j)$; $c_i = \cos\theta_i$; $c_{ij} = \cos(\theta_i + \theta_j)$, etc... and $x_{14} = g c_2 + h c_{23}$, $x'_{41} = -(g c_3 + h)$, letters f, h, g are the distances shown in

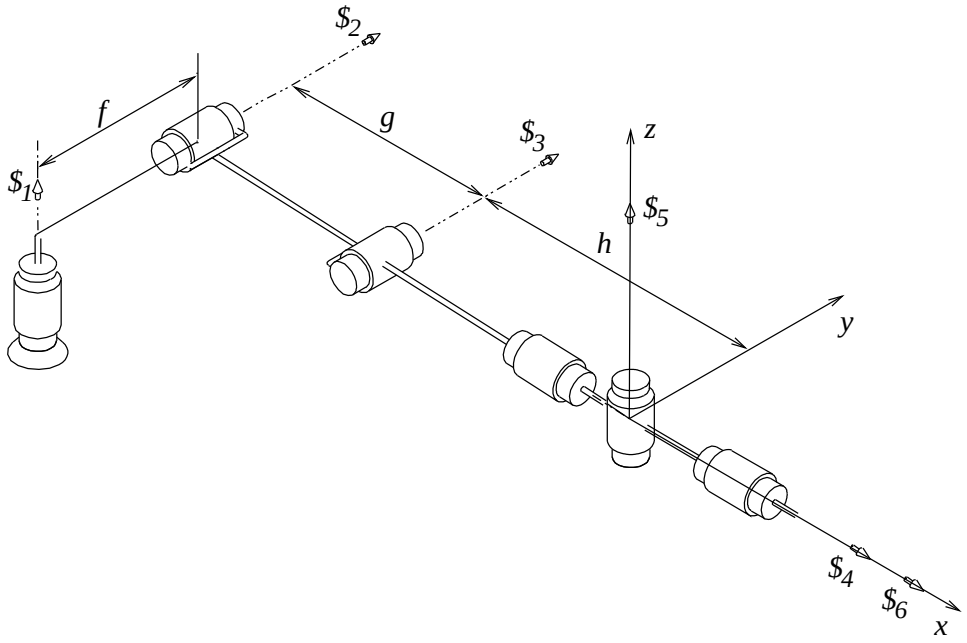

Figure 15: A sketch of the Puma robot with all joint variables at initial position.The coordinate system highlighted has origin at the wrist centre and is fixed on link 3.

Fig. 12. The structure of the inverse Jacobian is [Hunt 1987]

$$\overline{J^{-1}} = \begin{bmatrix} 0 & 0 & 0 & 0 & \times & 0 \\ 0 & 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & \times & \times & \times & \times & 0 \end{bmatrix} \tag{4.2}$$

where $\times$ represents the non-null terms of the matrix. The values of the inverse Jacobian is omitted here, as specific values of $\overline{J^{-1}}$ are not required by algorithm 1. The matrix $\overline{J^{-1}}$ is presented in [Hunt 1987], including some pertinent comments.

4.2 Hierarchical canonical form

We use the kinematic structure matrix $\bar{Q}$, eq. (2.16), to rearrange the system. The end-effector screw $\dot{x} = \$_n = [\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6]^T$ is represented in the coordinate system shown in Fig. 12, see details in [Hunt 1987].

Using Algorithm 1, the hierarchically rearranged linear system corresponding to eq. (2.16) is

$$\left[\begin{array}{c} \dot{x}_1 \\ \dot{q}_4 \\ \hline \dot{x}_2 \\ \dot{x}_3 \\ \dot{q}_5 \\ \dot{q}_6 \\ \hline \dot{x}_6 \\ \dot{q}_3 \\ \hline \dot{x}_4 \\ \dot{q}_2 \\ \hline \dot{x}_5 \\ \dot{q}_1 \end{array}\right] = \left[\begin{array}{cc|cc|cc|cc|cc|cc} & \times & & & & \times & & & & & & \times \\ \times & & \times & \times & & & \times & & \times & & \times & \\ \hline & & & & \times & \times & & \times & & \times & & \\ & & & & \times & \times & & & & & & \times \\ & & \times & \times & & & \times & & \times & & \times & \\ & & \times & \times & & & \times & & \times & & \times & \\ \hline & & & & & & & \times & & \times & & \times \\ & & & & & & \times & & \times & & \times & \\ \hline & & & & & & & & & \times & & \times \\ & & & & & & & & \times & & \times & \\ \hline & & & & & & & & & & & \times \\ & & & & & & & & & & \times & \end{array}\right] \left[\begin{array}{c} \dot{x}_1 \\ \dot{q}_4 \\ \hline \dot{x}_2 \\ \dot{x}_3 \\ \dot{q}_5 \\ \dot{q}_6 \\ \hline \dot{x}_6 \\ \dot{q}_3 \\ \hline \dot{x}_4 \\ \dot{q}_2 \\ \hline \dot{x}_5 \\ \dot{q}_1 \end{array}\right] \tag{4.3}$$

Extracting only the terms relative to the forward kinematics, *i.e.* dropping rows 2, 5, 6, 8, 10 and 12, eq. (4.3) becomes

$$\left[\begin{array}{c} \dot{x}_1 \\ \hline \dot{x}_2 \\ \dot{x}_3 \\ \hline \dot{x}_6 \\ \hline \dot{x}_4 \\ \hline \dot{x}_5 \end{array}\right] = \check{J} \left[\begin{array}{c} \dot{q}_4 \\ \hline \dot{q}_5 \\ \dot{q}_6 \\ \hline \dot{q}_3 \\ \hline \dot{q}_2 \\ \hline \dot{q}_1 \end{array}\right] \quad \text{where } \check{J} = \left[\begin{array}{c|cc|cc|c} 1 & & c_5 & & & -s_{23} \\ \hline & -s_4 & c_4 s_5 & 1 & 1 & \\ & c_4 & s_4 s_5 & & & c_{23} \\ \hline & & & -h & x'_{41} & -f s_{23} \\ \hline & & & & g s_3 & -f c_{23} \\ \hline & & & & & x_{14} \end{array}\right] \tag{4.4}$$

Matrix $\check{J}$ is the hierarchical canonical form of the Jacobian given by eq. (4.1). In the following text, we describe how this hierarchical canonical form can be

explored to solve the rearranged system, eq. (4.4), *i.e.* to invert the Jacobian by decomposition. In the sequence, we detail the inversion of $\check{J}$ as well the singularities of the robot architecture. Finally, section 4.5, we discuss other extras advantages that may be obtained by using this hierarchical canonical form.

4.3 Jacobian Inversion

Equation (4.4) permits the inversion via smaller subsystems, which cannot be performed directly on the original eq. (4.1). In other words, the hierarchical canonical form lightens the computational efforts and errors of inverting a higher dimensional matrix.

The process of inverting using subsystems takes the following steps: decomposition of the system into subsystems, ordering of the subsystems, use of variables obtained in prior steps and inversion of the diagonal block. Each subsystem is associated with only one diagonal block and corresponds to the rows where this diagonal block belongs to. The system, now decomposed, has to be solved sequentially. Bottom rows in eq. (4.4) have to be solved first, and upper rows have to wait for the variables obtained in the earlier subsystems. The method, in fact, is the same as the inversion of the Scara Jacobian using eqs. (1.5)-(1.7), but the Puma robot Jacobian, due to its complexity, turn the advantages of the method more explicit.

Once the system is subdivided and the subsystems are ordered, the process continues with the flow of variables from prior subsystems to following subsystems. An intermediate step waits for the availability of variables obtained in previous steps, and, once solved, it turns more variables accessible to the next steps. In this process, to solve a subsystem is basically to invert the diagonal block; when such inversion is not possible we can say that a singularity is reached.

We illustrate the process of the flow of variables, described above, solving the

subsystem comprised by the rows 2 and 3 of eq. (4.4)

$$\begin{bmatrix} \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -s_4 & c_4 s_5 \\ c_4 & s_4 s_5 \end{bmatrix} \begin{bmatrix} \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{q}_3 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{q}_2 + \begin{bmatrix} 0 \\ c_{23} \end{bmatrix} \dot{q}_1 \tag{4.5}$$

At this step, variables $\dot{q}_1, \dot{q}_2, \dot{q}_3$ must be *a priori* known, so the last three terms can be transferred to the first member of eq. (4.5). Finally, the variables $\dot{q}_5$ and $\dot{q}_6$ come from the inversion of the diagonal block

$$\begin{bmatrix} \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = \begin{bmatrix} -s_4 & c_4 s_5 \\ c_4 & s_4 s_5 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{q}_3 - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{q}_2 - \begin{bmatrix} 0 \\ c_{23} \end{bmatrix} \dot{q}_1 \right\} \tag{4.6}$$

The inversion via decomposition into subsystems implicitly underline a flow of variables from past to future steps. In eq. (4.5) this flow is:

$$\dot{q}_1, \dot{q}_2, \dot{q}_3 \rightarrow \begin{array}{c} \text{subsystem} \\ \text{rows 2 and 3} \end{array} \rightarrow \dot{q}_5, \dot{q}_6 \tag{4.7}$$

In the hierarchical analysis approach, variables are not obtained all together in a monolithic way, as *e.g.* inverting directly: $\dot{q} = J^{-1}\dot{x}$; in contrast, they are revealed gradually in small groups, one group at each step.

4.4 Hierarchical analysis

Hierarchical analysis is the kinematic analysis performed using the hierarchical canonical form of the Jacobian. First, we associate the hierarchical canonical form with the condensation digraph of the rearranged kinematic matrix $\check{Q}$. Second, using the graphical appeal of the condensation digraph derived from $\check{Q}$, we describe the flow of variables discussed above. Finally, at the end of the section, we discuss how each singularity affects the Puma robot.

The condensation digraph, Fig. 16, of the robot can be directly obtained from eq. (4.3), instead of the procedures in section 2.2. The horizontal lines divide the subsystems of the robot Jacobian. If we compare each subsystem described in Fig. 16, we notice that the flow of variables in the subsystem corresponds to the

flow of variables at the corresponding vertex of the condensation digraph. For instance, eq. (4.6) corresponds to vertex D in Fig. 16 which receives information from vertices A, B, C and transmits some information to vertex E. Considering only the joint variables, which are solved in the inverse kinematic problem, we return to eq. (4.7).

In a broader view, the condensation digraph maps the flow of variables all over the inversion process. The first variables to be solved correspond to the source variables of the condensation digraph, and the last variables to be solved correspond to the sink variables of the condensation digraph. Furthermore, each subsystem is, in fact, a strong component of the digraph. In conclusion, to invert the Jacobian in block-triangular form is equivalent to sweep all strong components in order, *i.e.* from source A to sink E in Fig. 16.

The flow of variables through the strong components induces a hierarchy of the diagonal blocks. When one of the these diagonal blocks loses rank the whole matrix J (or $\check{J}$) also loses rank, yet not all subsystems are affected by this singularity. The diagonal block which is not invertible when represented in the the condensation digraph is a vertex which cannot be reached nor trespassed.

The Puma robot has five diagonal blocks whose determinants are, from bottom to top : $x_{14}, gs_3, -h, -s_5, 1$. The terms $-h$ and 1 are invariant, *i.e.* do not depend upon the joint variables, and can be disregarded. The remaining terms: $x_{14}, gs_3, -s_5$, can lead to a singularity when they become null, so the Puma robot has 3 singularities: $x_{14} = 0, s_3 = 0, s_5 = 0$, which have different weight on the behaviour of the robot.

The first singularity, $x_{14} = 0$, affects all the hierarchical canonical form described in eq. (4.4) and can be considered the most drastic singularity of the Puma robot. The second singularity , $s_3 = 0$, is also quite severe; however, contrarily to the first singularity, the value of $\dot{q}_1$ is precisely determined in the inverse kinematics. In other words, the first step $\dot{q}_1 = (1/x_{14})\dot{x}_5$ was solved "before" the singularity "is computed". The last singularity, $s_5 = 0$, is the least severe singularity considering that it affects less variables than the previous ones.

For illustration purposes, we will describe the last singularity in more detail. When $s_5 = 0$ the condensation digraph can be cut into two parts, Fig. 16, one free from the singularity effects and the other affected by this singularity.

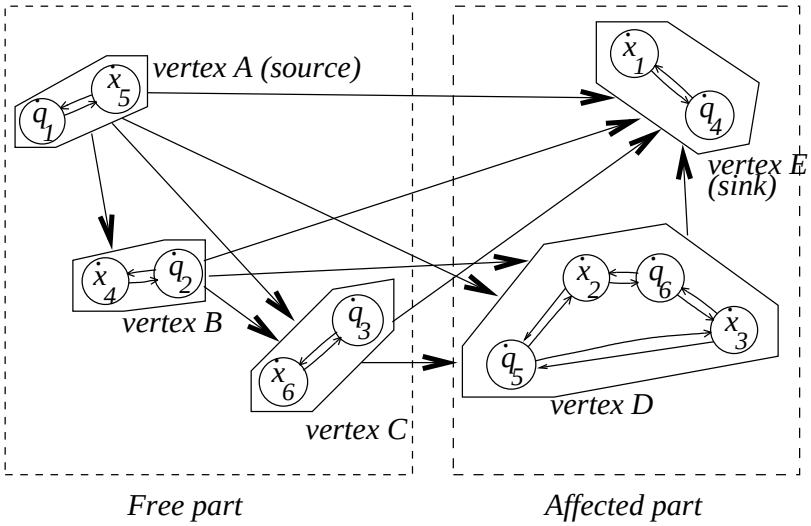

Figure 16: Condensation digraph of a Puma robot Jacobian based on the system given by eq. (4.3). Vertex A of the condensation digraph is the source of the digraph and vertex E is the sink of the digraph. A singularity at the vertex D divides the condensation digraph in two parts: one affected by the singularity and the other free from singularity effects

The affected part of Fig. 16 is composed by two types of variables: joint velocities $\dot{q}_i$ and end-effector directions $\dot{x}_i$. The joint velocities in the affected part are those velocities which tends to have inadmissible higher magnitudes nearby the singularity. On the other hand, end-effector directions in the affected part are those directions which will certainly have errors, because the corresponding joint velocities will not reach the high values required.

4.5 Other advantages

Above we described the two main advantages of the hierarchical analysis, but other advantages are also a consequence of the hierarchical canonical form.

We will present here another two incidental advantages: direct inversion and evaluation of the determinant.

Although the division of the whole system in smaller subsystems is a more rewarding inversion procedure, we can also point out that the *direct* inversion of $\check{J}$ is better than the direct inversion of J. We can notice that the shape of the system represented eq. (4.4) is improved when compared with eq. (4.1).

First, eq. (4.4) can be solved directly using methods such as LU decomposition [Lucas, Tischler e Samuel 2000] with better results than eq. (4.1). The linear system in eq. (4.4), in block-triangular form, has the pivoting process almost done beforehand. Moreover, when the matrix has to be pivoted, the choice of the pivot is limited to the elements of the diagonal blocks. Other point is that "fill-in" generation (elements that were null and becomes non-null, *i.e.* becomes filled in) is minimal when the matrix is its finest block-triangular form [Golub e Loan 1983], which requires less storage capability.

Secondly, the determinant is easier to obtain in eq. (4.4) than in eq. (4.1). Each diagonal block in eq. (4.4) for one term of the product which composes the determinant. For instance,

$$\det J = \det \check{J} = (x_{14}) \cdot (g s_3) \cdot (-h) \cdot \left(\det \begin{bmatrix} c_4 & -s_4 \\ s_4 & c_4 \end{bmatrix} \right) \cdot (1) \qquad (4.8)$$

5 *Hierarchical Kinematic Analysis of Redundant Robots*

This chapter presents the hierarchical kinematic analysis in the context of redundant manipulators. First, in section 5.1, the redundancy resolution is presented. Section 5.3 presents the extended Jacobian method, that is the most affordable method to use the hierarchical kinematic analysis. In section 5.3.1, this method is briefly compared with another important method in redundancy resolution, the generalised inverse (or pseudoinverse) method.

To our purposes, the extended Jacobian method, as originally conceived, is quite limited. Therefore, we propose a procedure for extending the Jacobian, *i.e.* how to select the extra lines that shall be added to the matrix. Section 5.4 describes this method, designated here as *globally constrained Jacobian method*. Despite this new method be based on the extension of the Jacobian, it has a completely different approach to the redundancy resolution. In section 5.5, this method is applied to the Roboturb manipulator.

5.1 Introduction to redundant robots

Robotic arms are defined as kinematically redundant when the number of joints is greater than the number of degrees of freedom needed to describe a specific task. The most general task of a manipulator is a general movement of the end-effector along the three-dimensional Cartesian space. This general

movement requires 6 degrees of freedom ($d = 6$) *i.e.* 6 independent variables to be completely determined at every configuration. Therefore, every robot with more than 6 independent joints is a redundant robot, as each joint is responsible for an extra independent variable to the system. In a planar case ($d = 3$), on the other hand, a manipulator with just four joints ($f = 4$) is always redundant no matter which task is ascribed to it. The difference $f - d$ is called the *degree of redundancy*.

The interest in redundant manipulators has growing in the last two decades. The main reason is that redundant manipulators can handle with movements in confined or cluttered spaces. Other reason is the extra ability that redundant manipulators have to avoid singularities, obstacles and joint limits. Another reason is that many specific applications are derived from, or may be significantly improved by, the use of redundant manipulators [Nenchev 1989]. We can cite *e.g.* multi-fingered hands, cooperative robots[1] [Hu e Goldenberg 1993], sheep shearing [Trevelyan, Kovesi e Ong 1983] [Trevelyan 1992], and manipulation in space [Hu e Vukovich 1998].

The first difficulty with redundant robots is in the design stage. Very few industrial robots are redundant, so most of the redundant robots must be specifically designed and manufactured by the interested part. At this stage, the designer must choose the degree of redundancy of the robot. Higher degree of redundancy has stronger disadvantages, such as higher weight, complexities in the control system, and higher cost. However, Hollerbach (1985) shows that a seven degrees of freedom manipulator can do the tasks described above. For this reason, unless otherwise stated, we will consider only manipulators with one degree of redundancy, $f - d = 1$, in this work.

[1] Two or more robots together cooperating in performing a common task

5.2 Redundancy resolution

After the path planning, the methods to deal with redundant and nonredundant manipulators differ significantly. The path planning is, in few words, a quest for a suitable path to perform the desired task. Once established in Cartesian space, the path has to be converted to the joint space via inverse kinematics. This step for a redundant manipulator is sometimes called the *redundancy resolution* of the manipulator, *i.e.* the computation of the joint variables, as accurately as possible, to follow the path previously defined. In general, the redundancy resolution yields an infinite number of joint solutions corresponding to a desired end-effector position/orientation. This multiplicity of solutions is the main difficulty of redundant manipulators. The redundancy resolution may be also described as the choice of one inverse kinematic solution among an infinitude of solutions.

Two approaches are suggested in the literature to the redundancy resolution: the use of generalised inverses, such as the Moore-Penrose pseudoinverse [Klein e Huang 1983], or the use of the extended Jacobian [Baillieul 1985] [Chang 1986] [Klein, Chu-Jenq e Ahmed 1995] *i.e.* a Jacobian with extra rows. The classical approaches cited above, namely the extended Jacobian and pseudoinverse methods, come from a minimisation procedure. For instance, the Moore-Penrose pseudoinverse method minimise the Euclidean norm of the vector composed by the joint variables, which is closely related to the minimisation of the kinetic energy [Siciliano 1993]. The extended Jacobian method impose extra constraints, but they are in general derived from a minimisation procedure.

The extended Jacobian method is based on adding more rows to the Jacobian, and is described in more detail in section 5.3. Section 5.3.1 presents a brief comparison between the extended Jacobian method and generalised inverses methods. A more detailed comparison can be found elsewhere [Klein, Chu-Jenq e Ahmed 1993], [Park, Chung e Youm 1994], [Baker e Wampler 1988]. A good review on the use of pseudoinverses for redundancy resolution is found in [Klein, Chu-Jenq e Ahmed 1995].

Section 5.4 describes the globally constrained Jacobian method, here proposed, based on the interpretation of the geometry and kinematics of the manipulator. The globally constrained Jacobian method is also based on an additional row to the Jacobian matrix in order to invert it. In this sense, the globally constrained Jacobian method has close similarity with the extended Jacobian method; however, both methods have different formulations and philosophies as will be described in the next sections.

5.3 Extended Jacobian method

The extended Jacobian method is based on imposing additional constraint functions of the form $h(q) = 0$. The proponents of this method, Baillieul (1985) and Chang (1986), use this extra constraint as a secondary task to be simultaneously satisfied. They also give formulations for converting minimisation criteria into these constraint functions.

Once the functions $h(q) = 0$ are determined, one must solve simultaneously the combined set of equations

$$\begin{aligned} x(t) &= f(q(t)) \\ 0 &= h(q) \end{aligned} \tag{5.1}$$

where $f(q(t))$ represents the function which relates the joint positions $q(t)$ and the Cartesian positions of the end-effector $x(t)$. Equations (5.1) can be solved numerically at each point, as suggested by [Chang 1986] or, when h and f are differentiable, an initial solution to eq. (5.1) can be propagated along a path by solving the differential kinematic equation

$$J_e \dot{q} = \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \tag{5.2}$$

where J_e was named by Baillieul (1985) as the *extended Jacobian matrix* defined by

$$J_e = \begin{bmatrix} J(q) \\ \partial h(q)/\partial q \end{bmatrix} \tag{5.3}$$

The condition for the existence of a unique solution $\dot{q}$ in eq. (5.2) is that the extended Jacobian matrix J_e be nonsingular along the entire path. General spatial manipulators with one degree of redundancy have $J(q)$ as a 6×7 matrix, $J_{6\times7}$, and J_e is a 7×7 matrix. In this case $h(q)$ is a scalar function of vector q. Unless otherwise stated, this is the degree of redundancy that we consider in this chapter; nevertheless, all methods described here are, *mutatis mutandis*, applicable to higher degrees of redundancy.

Klein *et al.*(1993) asserts that the presence of algorithmic singularities is one of the disadvantages of the method. Algorithmic singularities are those created by the presence of the extra row and which were not present at the original Jacobian $J_{6\times7}$. From eqs. (5.2) and (5.3) we noticed that $\det J_e = 0$ can have solutions which are not related to the kinematics of the robot itself and are purely related to the "intrusion" of the extra row $\partial h(q)/\partial q$. In this sense, the advantages of redundant robots cited above may be limited by some spurious algorithmic singularities, possibly generated by a bad selection of the constraint function $h(q)$.

The choice of the "objective criterion $h(q)$" is described by Klein (1995) as a rather complex task. The literature provides general objective criteria $h(q)$ which tend to be more appropriate to one or another problem, such as the avoidance of obstacles, singularities or joint limits.

The most common objective criterion is [Klein, Chu-Jenq e Ahmed 1995]

$$h(q) = \sum_{i=1}^{n} q_i^2 \tag{5.4}$$

Equation (5.4) is called the *joint-range availability* criterion because it tends to keep the joints near their centre of travel and away from mechanical limits[2]. The

[2]We are considering that the joint angles origin are set at the centre of the travel of the

last row of eq. (5.3) will be then equal to $2\,[q_1\ q_2\ \cdots\ q_n] = 2q$

The equation which corresponds to the last row in eq. (5.2) is

$$\partial h(q)/\partial q = 0 \tag{5.5}$$

Dividing eq. (5.5) by 2 and appending it to the original Jacobian generates the extended Jacobian:

$$J_e = \begin{bmatrix} J_{6\times 7}(q) \\ q_1\ q_2\ \cdots\ q_7 \end{bmatrix} \tag{5.6}$$

The presence of all joint variables in the last row of J_e is a clever way to avoid selecting which variables are important to the performance of the robot and which variables are not; however, the full last row increases the presence of unexpected algorithmic singularities. Moreover, the full last row couples the associated digraph of eq. (5.6) perhaps more than necessary.

Despite these facts, the use of the extended Jacobian method has important advantages over the pseudoinverse methods, and a comparison of both methods will be presented next.

5.3.1 Comparison with generalised inverse methods

The generalised inverse methods are based on a minimisation of a criterion to choose the best option, according to this criterion, among the infinitude of possible solutions. Several generalised inverse methods exist [Doty, Melchiorri e Bonivento 1993]; however, the most used generalised inverse is the Moore-Penrose pseudoinverse or, for short, the pseudoinverse. This section will present some comparative aspects of these methods in relation to the extended Jacobian method.

The extended Jacobian method is claimed by some authors [Klein, Chu-Jenq e Ahmed 1995] as being easier to set. The main goal of all these methods is to

joint. If this is not the case a simple adjustment can be done on eq. (5.4)

obtain a good performance between a sluggish manipulator and an unstable one. In this aspect, the selection of an additional constraint can be easier than the selection of suitable weight factors in the generalised inverse methods.

Other aspect is that the extended Jacobian method provides, under certain conditions, a *conservative movement*, *i.e.* one closed path on the end-effector space causes a closed path on the joint variables space. Baker & Wampler (1988) demonstrate that the pseudoinverse methods cannot obtain conservative movement while extended Jacobian methods guarantee conservative movement within a simply connected region without internal singularities. This fact is reinforced in simulations and experiments elsewhere [Nenchev 1989] [Klein e Huang 1983].

Similarly to the pseudoinverse solution, the extended Jacobian method, as it was originally proposed, can be considered as a particular case of other formulations. For example, Nenchev (1989) shows that the extended Jacobian method is a particular case of the constrained least-square method. In reality, both approaches are closely intertwined. For instance, the problem of algorithmic singularities, which is characteristic of extended Jacobian methods, is encountered also in pseudoinverse-based solutions, such as the task-priority redundancy resolution discussed in [Chiaverini 1997].

However, the difficulties of selecting precisely the function $h(q)$, the coupling of the associated digraph and the presence of unexpected algorithmic singularities still remain. The selection of a suitable function $h(q)$ at positional level is not simple. General criteria, like the joint availability criterion of eq. (5.4), cannot be widely applied. Most robot architectures demand for a specific function $h(q)$.

These facts lead us to reconsider the extended Jacobian method. For these reasons, we developed a method also based on the extension of the Jacobian but with a different methodological perspective. To distinguish this method we designate this variant as the *globally constrained Jacobian method* while keeping the designation "extended Jacobian method" to the classical methods found in the literature up to now. In fact, the globally constrained Jacobian method is an "extended Jacobian method". Their differences will be listed in the section 5.4.

5.3.2 Hierarchical kinematic analysis using extended Jacobian method

As described above, the full last row of the classical extended Jacobian method couples the associated digraph of eq. (5.6) perhaps more than necessary. This strong coupling has two negative effects: singularities have a more complex expression and are harder to predict and to avoid, and the use of hierarchical kinematic analysis becomes less beneficial.

5.4 Globally constrained Jacobian method

The globally constrained Jacobian method, as the extended Jacobian method, is a method based on appending an extra row to the Jacobian matrix; therefore, avoiding generalised inverses and permitting hierarchical kinematic analysis. The constraint, in this case, is not imposed considering a minimisation process or the robot function $h(q)$ at a positional level alone; instead, the extra row is selected based on the task and on the geometry of the manipulator.

The crucial difference between both methods emulates the technical and philosophical conflict between the choice of general methods in Kinematics, *i.e.* the methods which can be applied to a wide class of mechanical architectures, and particular methods, *i.e.* those fitted to (optimally) satisfy one specific architecture and whose adaptation to another mechanism involves redesigns [Fanguella e Galletti 1989]. In our case, the classical extended Jacobian method provides a global kinematic solution to the problem while the globally constrained Jacobian method provides a specific solution to each robot architecture.

The choice of this extra row in the globally constrained Jacobian method is not based on an assumed position function, but rather on a *velocity constraint*. We will illustrate the concept with an example: a redundant seven degrees-of-

freedom robot. This kind of manipulators has a Jacobian matrix such as

$$J_{6\times 7} = \begin{bmatrix} \$_1 & \$_2 & \cdots & \$_7 \end{bmatrix} \tag{5.7}$$

where each one of the seven columns is the 6 Plücker's coordinates of the screw $\$_i$ corresponding to the joint variable q_i [Hunt 1987]. The equation which corresponds to the last row takes the general form

$$r = \vec{n}\dot{q} \tag{5.8}$$

The variable r is a scalar, the *redundancy parameter*, which maps the "movement" of the *constraint plane* whose normal is given by $\vec{n}$. Vector $\vec{n}$ can be *e.g.* $\vec{n} = e_1 = [1\ 0\ 0\ 0\ 0\ 0\ 0]$ which reduces eq. (5.8) to a simpler form: $r = \dot{q}_1$. In this case, the redundancy resolution is made by keeping control only of the velocity of the first joint. For example, this velocity can be null; therefore, the first joint is blocked and the remaining 6 joints are solved using the traditional form of the inverse kinematics.

Comparing the extended row of the globally constrained Jacobian method, eq. (5.8), and the extended row of the classical extended Jacobian method, eq. (5.5), an important difference is perceived. Equation (5.5) imposes a constraint which is a hyperplane passing necessarily through the origin of the joint space $\mathbb{R}^7$ spanned by the joint variables $q_1, q_2, q_3, q_4, q_5, q_6, q_7$.

Figure 17 shows a projection of the joint space $\mathbb{R}^7$ into a 2-dimensional plane corresponding to two generic variables q_i and q_j. Each inclined line corresponds to the projected constraint. Distances $\bar{r}_i$ are projections in this two-dimensional subspace of some values of the redundancy parameter r.

The extended Jacobian method has no flexibility to vary r (and correspondingly $\bar{r}_i$). In the extended Jacobian method the hyperplane representing the constraint passes obligatorily through the origin of $\mathbb{R}^7$. In this case, the redundancy parameter does not exist ($r \equiv 0$) and the following simple kinematic interpretation is not feasible.

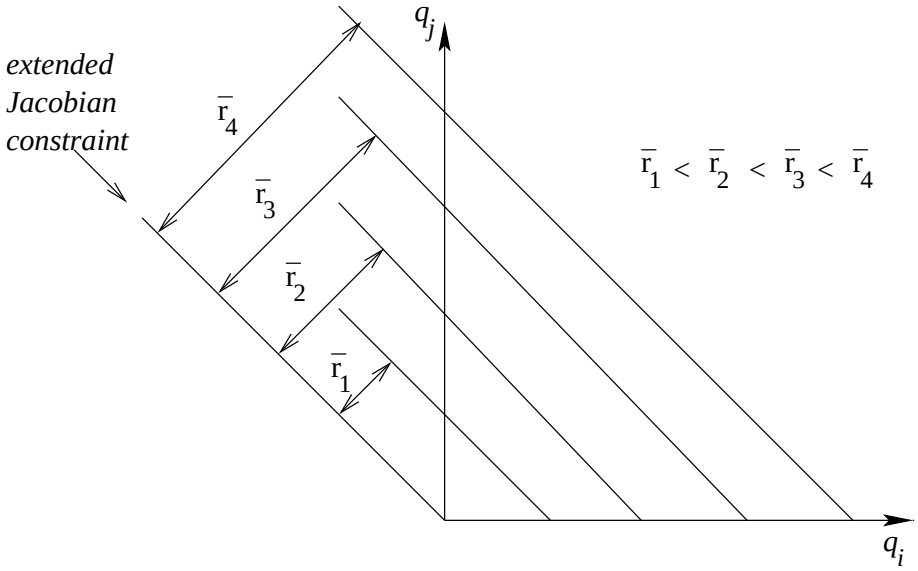

Figure 17: Comparison of the constraints of the globally constraint Jacobian method and the classical extended Jacobian method seen from a 2-dimensional projection of the $\mathbb{R}^7$ space. Scalars $\bar{r}_i$ are projections of the values of r in eq. (5.8). Negative values for r, and correspondingly $\bar{r}_i$ are also possible.

5.4.1 The simplest choice $\vec{n} = e_i$

In fact, the simple choice $\vec{n} = e_i$ can be suitable when the consequences of this choice are known. In the remaining of this work we will adopt solutions of the form $\vec{n} = e_i$ explaining the reasons which lead us to the choice. However, this is not the unique solution to the problem.

The above case leads to

$$\vec{n} = e_i \Longrightarrow r = \dot{q}_i \tag{5.9}$$

which provides a strategy to avoid singularities. The Roboturb robot can serve as an example. The Roboturb robot has three singularities, see details in section 5.5 and [Martins e Guenther 2001]. Two of these singularities are $s_4 = 0$ and $s_6 = 0$ which means that when $q_4, q_6 = 0$ (or π) the Jacobian matrix drops rank.

The proximity of a singularity can be monitored by the functions s_4 and s_6. For example, when $|s_4|$ is near zero some joint velocities will become exceedingly

large. Keeping $|s_4| > \epsilon_4$ can avoid such situation, here ϵ_4 is a scalar which serves as a threshold chosen to obtain a certain degree of velocities around the singularity. When this threshold is reached, the last row can be changed to a more appropriate constraint like $\vec{n} = e_4 = [0\ 0\ 0\ 1\ 0\ 0\ 0], \Longrightarrow r = \dot{q}_4$. As the manipulator approaches an unsafe zone, the performance of the robot decreases. A strategy to avoid this situation must be given. A way to control the redundancy parameter is by imposing

$$r = -k\frac{\overline{\dot{q}_4}}{\|\overline{\dot{q}_4}\|}, \qquad k > 0 \tag{5.10}$$

where $\overline{\dot{q}_4}$ is the last velocity which the manipulator reached the subspace of the Cartesian space which corresponds to the ball $\{q \in \mathbb{R}^n | \dot{q}_4 \leq \epsilon_4$ in the joint space. Details of this method of singularity avoidance can be found in [Martins, Simas e Guenther 2001].

5.4.2 Algorithmic singularities

Algorithmic singularities are one of the problems which is derived from the extended Jacobian method and some authors, *e.g.* [Klein, Chu-Jenq e Ahmed 1993], point out that this is the most critical problems with this method.

The globally constrained Jacobian method may also present algorithmic singularities but, as the added row is normally independent of the joint variables, the mapping of these algorithmic singularities becomes quite easier.

Let us consider an added row in the form $\vec{n} = e_i$. In this case the algorithmic singularities are avoided when the minor determinant excluding the added row and the i-th column is kept non-null. This submatrix naturally will drop rank only when the associated non-redundant robot, *i.e.* the robot with the $i-$joint fixed, reaches a singular configuration. This type of singularity can be considered troublesome, but it cannot be considered unexpected as it has a clear connection with the robot geometry.

5.4.3 Hierarchical kinematic analysis using globally constrained Jacobian method

The globally constrained Jacobian method has the same features of the extended Jacobian method, as both methods rely on the the extension of the Jacobian. However, as the design of this extra row has no direct relationship with a positional constrained, the freedom of choice of this line is greater.

In the globally constrained Jacobian method, the extra row has normally two basic characteristics: the row is quite simple such as $\vec{n} = e_i$, and normally is independent of the positional parameters $\vec{n} \neq \vec{n}(q)$.

Both characteristics lead to a minimal coupling of the associated graph of J_e and yield a better hierarchical kinematic analysis. The rearranged system has smaller diagonal blocks, each one related to a unique singularity. The overlapping of singularities in strongly avoided at this stage, whenever the robot architecture permits, and unexpected singularities are minimised.

5.5 Hierarchy of the singularities of the Roboturb robot

The Roboturb robot is a serial manipulator with seven degrees of freedom originally conceived to repair turbine blades [Guenther, Simas e Pieri 2000]. The first joint is prismatic, and the remaining 6 joints are rotative. The last three joint axes intersect one another in a single point forming a so-called spherical wrist. Figure 18 illustrates the manipulator that will be analysed in this work.

Figure 19 shows a sketch of the manipulator architecture, and the chosen coordinate system where screw theory methods are applied. This Figure remarks the initial configuration of the robot *i.e.* $q_i = 0 \forall i$.

Screws $\$_i$ at joints are shown in Fig. 19. Position angles q_i are associated with each joint of the manipulator and have positive orientation shown in Fig. 19.

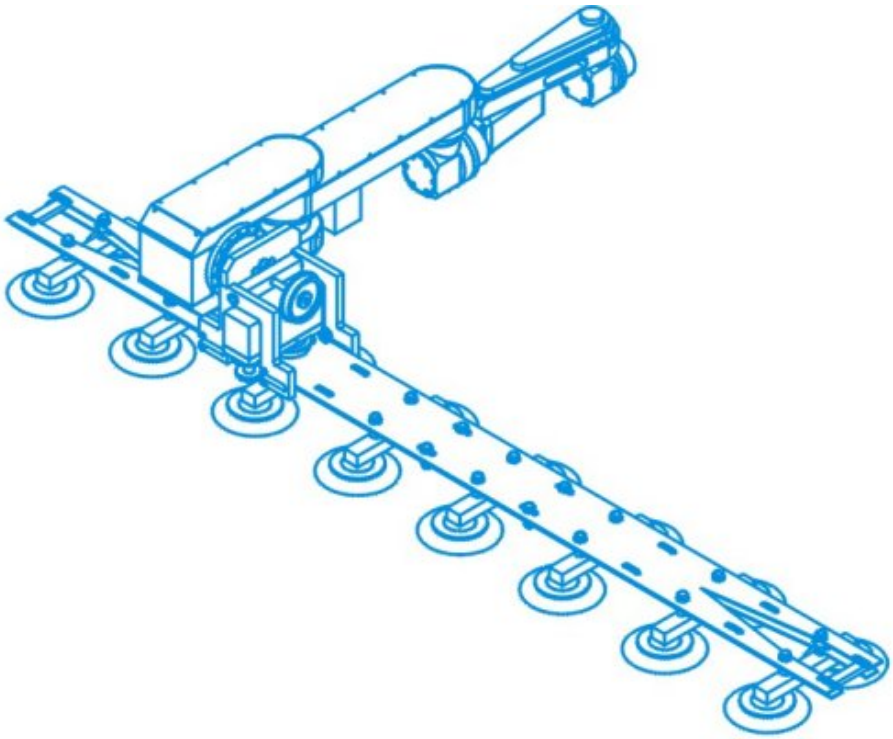

Figure 18: The Roboturb manipulator

The first joint, represented by $\$_1$, is prismatic and has translational displacement along the direction of $\$_1$. The chosen coordinate system is represented by the triad x, y, z fixed on link 4. Figure 19 shows the coordinate system chosen and the manipulator at the null position, *i.e.* at the position where all joint variables are zero, $q_i = 0$ for $i = 1, 2, \cdots, 7$.

The Jacobian corresponding to Fig. 18 is

$$J = \begin{bmatrix} 0 & -s_{34} & 0 & 0 & 1 & 0 & c_6 \\ 0 & -c_{34} & 0 & 0 & 0 & -s_5 & c_5 s_6 \\ 0 & 0 & 1 & 1 & 0 & c_5 & s_5 s_6 \\ -s_{34} & d_4 c_{34} & a_2 s_4 & 0 & 0 & 0 & 0 \\ -c_{34} & d_4 c_{34} & x_{24} & a_3 & 0 & 0 & 0 \\ 0 & x_{14} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.11}$$

where $s_i = \sin q_i$; $s_{ij} = \sin(q_i + q_j)$; $c_i = \cos q_i$; $c_{ij} = \cos(q_i + q_j)$, etc ... The

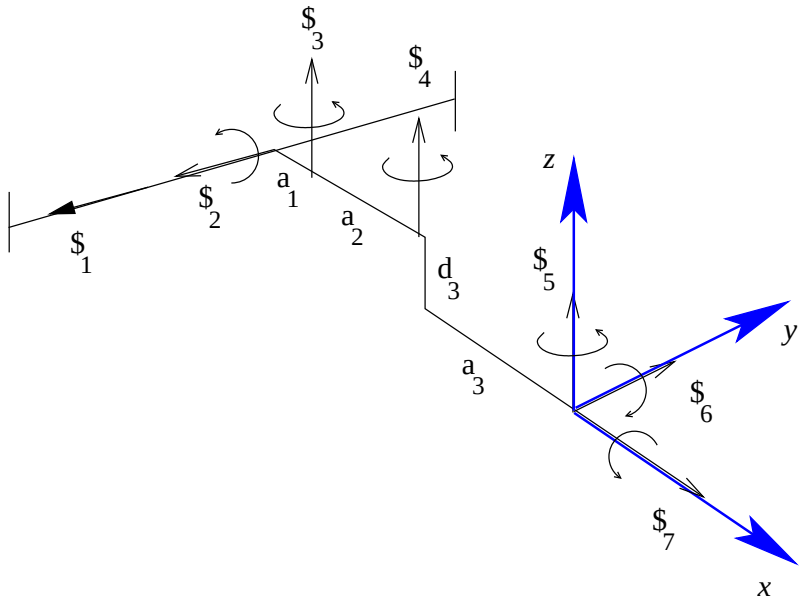

Figure 19: Coordinate system chosen to the Roboturb robot. The coordinate system is shown at the initial configuration *i.e.* the configuration where all the joint variables q_i are null. Notice that $\$_1$ corresponds to the prismatic joint displacement along the trail while the remaining six screws ($\$_1 - \$_7$) are rotational displacements.

symbols a_1, a_2, a_3 are distances shown in Fig. 19.

Here $x_{14} = a_1 + a_2c_3 + a_3c_{34}$ and $x_{24} = a_3 + a_2c_4$ represent distances along the x-axis from the spherical wrist centre to joints 1 and 2 axles, respectively. This notation, based on [Hunt 1987], was also used for other Jacobians such as the Puma robot.

The Jacobian of eq. (5.11) is extended by an extra row $\vec{n} = e_1$, as the main positional movement of this manipulator is performed over the prismatic joint at the first joint. This decision is in accordance with the kinematic conception of the robot described in [Guenther, Simas e Pieri 2000]. This redundancy resolution may be interpreted as: *given the desired velocity of the manipulator over the trail,* $\dot{q}_1(= r)$*, the system is solved for the remaining velocities through inverse*

kinematics. This results in the extended Jacobian:

$$J_e = \left[\begin{array}{ccccccc} 0 & -s_{34} & 0 & 0 & 1 & 0 & c_6 \\ 0 & -c_{34} & 0 & 0 & 0 & -s_5 & c_5 s_6 \\ 0 & 0 & 1 & 1 & 0 & c_5 & s_5 s_6 \\ -s_{34} & d_4 c_{34} & a_2 s_4 & 0 & 0 & 0 & 0 \\ -c_{34} & d_4 c_{34} & x_{24} & a_3 & 0 & 0 & 0 \\ 0 & x_{14} & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{5.12}$$

The structure of the inverse of the extended Jacobian is obtained numerically using algorithm described in the appendix D

$$\overline{J_e^{-1}} = \left[\begin{array}{ccccccc} 0 & 0 & 0 & \times & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{array}\right] \tag{5.13}$$

The hierarchically arranged linear system is

$$\begin{bmatrix} \omega_x \\ \dot{q}_5 \\ \hline \omega_y \\ \omega_z \\ \dot{q}_6 \\ \dot{q}_7 \\ \hline v_y \\ \dot{q}_4 \\ \hline v_x \\ \dot{q}_3 \\ \hline v_z \\ \dot{q}_2 \\ \hline r \\ \dot{q}_1 \end{bmatrix} = \left[\begin{array}{cc|cc|cc|c} \times & & & \times & & & \times & \\ \times & & \times\times & & \times & \times & \times & \times \\ \hline & & & \times\times & & & \times & \\ & & & \times\times & \times & \times & & \\ & & \times\times & & \times & \times & \times & \times \\ & & \times\times & & \times & \times & \times & \times \\ \hline & & & & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times \\ \hline & & & & & \times & \times & \times \\ & & & & & \times & \times & \times \\ \hline & & & & & & \times & \\ & & & & & & \times & \\ \hline & & & & & & & \times \\ & & & & & & & \times \end{array}\right] \begin{bmatrix} \omega_x \\ \dot{q}_5 \\ \hline \omega_y \\ \omega_z \\ \dot{q}_6 \\ \dot{q}_7 \\ \hline v_y \\ \dot{q}_4 \\ \hline v_x \\ \dot{q}_3 \\ \hline v_z \\ \dot{q}_2 \\ \hline r \\ \dot{q}_1 \end{bmatrix} \tag{5.14}$$

where the end-effector screw $\dot{x} = \$_n = [\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]^T$ is represented in coordinate system 4, see Fig. 19.

The hierarchically arranged linear system corresponding to eq. (5.14) is obtained using the techniques described in Chapter 2.

Extracting only the terms relative to the forward kinematics

$$\begin{bmatrix} \omega_x \\ \hline \omega_y \\ \omega_z \\ \hline v_y \\ \hline v_x \\ \hline v_z \\ \hline r \end{bmatrix} = \check{J}_e \begin{bmatrix} \dot{q}_5 \\ \hline \dot{q}_6 \\ \dot{q}_7 \\ \hline \dot{q}_4 \\ \hline \dot{q}_3 \\ \hline \dot{q}_2 \\ \hline \dot{q}_1 \end{bmatrix} \text{ where } \check{J}_e = \left[\begin{array}{ccc|ccc|c} 1 & & c_6 & & & -s_{34} & \\ \hline & -s_5 & c_5 s_6 & & & & \\ & c_5 & s_5 s_6 & 1 & 1 & & \\ \hline & & & a_3 & x_{24} & -d_4 s_{34} & -c_{34} \\ \hline & & & & a_2 s_4 & d_4 c_{34} & -s_{34} \\ \hline & & & & & x_{14} & \\ \hline & & & & & & 1 \end{array}\right] \tag{5.15}$$

Equation (5.15) has a strictly upper block-triangular form. The only 2×2 diagonal block has determinant $-s_6$ which is associated to the so-called wrist singularity of the manipulator. From bottom to top the hierarchy of special configurations is $(x_{14} = 0), (a_2 s_4 = 0), (-s_6 = 0)$. For instance $x_{14} = 0$ affects all screw coordinates of the end-effector while $s_6 = 0$ affects only $\omega_x, \omega_y, \omega_z$. The linear velocity components v_y, v_x, v_z of the end-effector are uniquely determined *even if* $s_6 = 0$. Physically, the spherical wrist singularity $(s_6 = 0)$ affects the orientation of the end-effector but not its position nor linear velocity.

These facts become more evident when we extract only the terms relative to the inverse kinematics

$$\begin{bmatrix} \dot{q}_5 \\ \hline \dot{q}_6 \\ \dot{q}_7 \\ \hline \dot{q}_4 \\ \hline \dot{q}_3 \\ \hline \dot{q}_2 \\ \hline \dot{q}_1 \end{bmatrix} = \widetilde{J_e^{-1}} \begin{bmatrix} \omega_x \\ \hline \omega_y \\ \omega_z \\ \hline v_y \\ \hline v_x \\ \hline v_z \\ \hline r \end{bmatrix} \tag{5.16}$$

where

$$\widetilde{J_e^{-1}} = \left[\begin{array}{c|cc|c|c|c|c} 1 & \times & \times & \times & \times & \times & \times \\ \hline & -s_5 & c_5 & \times & \times & \times & \times \\ & c_5/s_6 & s_5/s_6 & \times & \times & \times & \times \\ \hline & & & 1/a_3 & \times & \times & \times \\ \hline & & & & 1/(a_2 s_4) & \times & \times \\ \hline & & & & & 1/x_{14} & \\ \hline & & & & & & 1 \end{array}\right] \tag{5.17}$$

These results are quite similar to those obtained in [Martins e Guenther 2001] where the Puma robot in the coordinate system of Hunt (1987) is analysed. In fact, the Roboturb has an architecture similar to the Puma robot, but with the shoulder eccentricity located at a different place. The Roboturb manipulator has d_3 located between joints 3 and 4, while the Puma manipulator has f (or d_3

according to the Denavit-Hartenberg convention) located between joints 1 and 4 axes.

5.6 Final comments

This chapter reinforced that the hierarchical kinematic analysis can be successfully applied to redundant manipulators. The core of the problem is the use of an extended Jacobian and then using Algorithm 1 to reorder the extended (and square) Jacobian.

General positional criteria, such as the joint availability criterion of eq. (5.4), couples the associated digraph of eq. (5.6) more than necessary and makes the use hierarchical kinematic analysis less beneficial.

While the hierarchical kinematic method can be applied to any extended Jacobian method, the results can be quite deceptive when the positional criteria tend to couple all, or almost all, of the kinematic variables. The consequences of this coupling are *per se* negative, *e.g.* increasing the presence of unexpected singularities. For this reason, we provided a variant of the extended Jacobian method which minimise this coupling and, as consequence, yields a more useful hierarchical kinematic analysis.

At last, we observe that the obtained results, for the Roboturb manipulator, are quite similar to those obtained in chapter 4 and in [Martins e Guenther 2001], for the Puma robot, due to the similarity between both architectures.

6 Hierarchical Kinematic Analysis of Parallel Robots

This chapter deals with parallel robots and how the hierarchical kinematic analysis may contribute to the kinematic solution of this kind of manipulators. First, we describe the main differences between serial and parallel robots. In the sequence, we present the steps needed to apply the hierarchical kinematic analysis. Finally, the method is applied to a planar variant of the Stewart-Gough platform.

Parallel robots have, in general, simpler inverse kinematics than direct kinematics. Other two differences between parallel robots and serial robots require special attention before using the hierarchical kinematic analysis. First, singularities of parallel robots are severer than the singularities of serial robots. Second, the Jacobian matrix generally is not explicitly represented by the joint variables; it is numerically obtained from internal and external (dot and cross) products of vectors in the Cartesian space [Tsai 1999] [Valdiero et al. 2001].

Other characteristic of parallel robots is the presence of two types of singularities: singularities which diminish the degrees of freedom of the manipulator, and singularities which enlarge the degrees of freedom of the manipulator. Singularities in parallel robots are, in general, harder to predict and harder to avoid. As the Jacobian matrix is numerically obtained, most of these singularities can be even unexpected, so yielding hazardous consequences to on-line control.

6.1 Overview of the chapter

In this chapter, we apply the hierarchical kinematic analysis to parallel robots. Section 6.2 lists the prerequisites of the hierarchical kinematic analysis in order to direct the focus of the next sections. These prerequisites were intrinsically satisfied by the formulation applied to serial and redundant robots; however, standard parallel robot kinematics algorithms do not satisfy these prerequisites. Section 6.3 gives an overview of the structure of parallel robots and a brief history of these manipulators, with special attention to Stewart-Gough platforms on section 6.3.1. Section 6.5 describes the use of the Kirchhoff-Davies laws to obtain the parallel robots kinematics. Section 6.6 shows how to improve the solution of the linear system by using the hierarchical kinematic analysis. Section 6.8 shows the close relationship between the hierarchical kinematic analysis of parallel robots and their intrinsic decomposition into minimal Assur groups.

The first part of this chapter is based on the works of Davies (1981, 1983, 1995, 2000). The application of the Kirchhoff-Davies laws to the 3RRR planar platform in this chapter, section 6.5, and in Appendix E follows closely [Campos 2001]. The Assur group theory, section 6.8, has several independent sources that are specifically cited in section 6.8.

6.2 Requisites of the hierarchical kinematic analysis

Although not stated clearly in the earlier chapters, the hierarchical kinematic analysis has two basic requisites:

- A linear system involving input and output velocities, *e.g.* in the form $\dot{x} = J\dot{q}$
- The elements of the matrix in this linear system must be explicitly known for every configurations in space, *i.e.* $J = J(q)$ or $J = J(x)$.

The formalism used for serial robots in chapter 4 and the extension to redundant robots in chapter 5 underline both of these requisites. Chapter 3 derives the explicit relationship between each columns of the Jacobian matrix and the joint variables, see for instance Fig. 11 at page. 36. For parallel robots, a methodology that encompasses both requisites is not widely used, as almost all of the forms to obtain the Jacobian matrix are by internal and external (dot and cross) products of vectors in the Cartesian space. An explicit (functional) dependence relationship of the joint variables, or of the Cartesian coordinates, is not given.

The only method with attains both requisites seems to be the Kirchhoff-Davies laws [Davies 1983] [Davies 1981] [Davies 1995] [Davies 2000] [Campos 2001].

The Kirchhoff-Davies laws, however, do not lead to a Jacobian matrix, at least not directly. The method yields a linear system at a preliminary step in the form $[\hat{M}_N][\Psi] = [0]$ that defines completely the (closed) kinematic chain but not a particular mechanism. This mechanism is only defined when the actuators are selected. The Kirchhoff-Davies laws is described in more detail in Appendix E. A short description of these laws is in section 6.5.

6.3 Parallel robots: an introduction

Parallel manipulators differ from the more traditional serial robotic manipulators by virtue of their kinematic structure. Parallel manipulators are composed by one or more closed kinematic loops. Typically, these kinematic loops are formed by two or more kinematic chains that connect a moving platform to a base, where some of the joints of these kinematic chains does not have an actuator *i.e.* they are *passive*. This kinematic structure allows parallel manipulators to be driven by actuators positioned on, or near, the base of the manipulator [Tahmasebi 1992].

In contrast, serial manipulators, as those described earlier in this work, do not have closed kinematic loops, and are usually actuated at each joint along the serial linkage. In a serial chain, a significant part of the loads on the robot (including

link weights) is carried out by the actuators. The closer the base, the higher the load the actuator has to overcome. In this sense, the first actuators have to be more robust than the others. Reinforcing this fact, Trevelyan (1992) provides a simple rule of thumb: each extra joint added to a serial robot approximately doubles the weight of the manipulator.

Parallel manipulators distribute more evenly the loads among the actuators, so they can weight less than serial manipulators. Hence, parallel manipulators share the potential benefits associated with light weight construction, such as high-speed operation and improved load to weight ratios. On the other hand, parallel manipulators have smaller workspaces than serial manipulators of similar size.

Tahmasebi (1992) describes the main advantages of parallel robots as

- Good rigidity
- Simple inverse kinematics
- High payload capacity

Nevertheless, they have the following extra disadvantages:

- Position and orientation of the moving platform are highly coupled.
- Their direct kinematics are difficult to solve.
- Most parallel robots require precise spherical joints, which are difficult to manufacture at low cost.

Parallel robots architectures are usually of the platform-type, such as the Stewart-Gough platform [Gough e Whitehall 1962] [Stewart 1966]. These parallel robots are normally selected when serial robots are not suitable to the task. Two reasons justify this choice: their generic applicability, and the avoidance of the

design process of a proper parallel robot to perform the task, as the design of closed-chain mechanisms is not easy [Tischler, Samuel e Hunt 1995] [Tischler 1995] [Tischler, Samuel e Hunt 1998] [Tischler, Hunt e Samuel 1998]

Platform-type parallel robots are minimal kinematic chains, see details in [Tischler, Samuel e Hunt 1995] [Tischler 1995], and for this reason are complex when compared with non-minimal kinematic chains with the same number of links. Non-minimal kinematic chains may be easier to build and be more appropriate to a specific task. However, non-minimal kinematic chains require a previous design. The designer needs firm grounds on kinematics and geometry as well as some insight and patience to do this task.

6.3.1 Stewart-Gough platforms

Stewart (1966) proposes a mechanical architecture for a flight simulator, whose variants are still used in Aeronautics. This mechanism was based on the tyre testing machine devised by Gough & Whitehall (1962), *apud* [Tsai 2001]. Since then, the Stewart-Gough platform has also been applied in other areas, such as milling machines, pointing devices, and underground excavation equipments. Tsai (2001) provides more information at chapter 9 and references therein.

The Stewart-Gough platform has been extensively studied [Zhang e Song 1992] [Hunt e McAree 1998] [Ku 1999]. Generally, the Stewart platform has six limbs, where each limb is connected to both the base and the moving platform by spherical joints located at each end of the limb. The platform is typically actuated by changing the lengths of the limbs.

Moreover, the closed-form direct kinematic solutions that have been reported in the literature have generally required special forms of the Stewart-Gough platform [Parenti-Castelli e Gregorio 1996] [Zhang e Song 1992]. In these special forms, pairs of spherical joints may present design and manufacturing problems. Some of these special forms require concentric spherical joints and a harder design process, see comments on [Hunt e McAree 1998].

Due to the difficulties cited above, some of the current research focus on planar parallel manipulators, specially on the planar variants of the Stewart platform, *e.g.* [Wu e Huang 1996]. To illustrate the problem, we will exemplify the hierarchical kinematic analysis of a planar mechanism: the 3RRR planar platform. Nevertheless, the method described in the following sections may be generally applied to any other planar, spherical or spatial mechanism.

6.4 Kinematic problem of parallel robots

A serial robot has, in general, no passive joints, *i.e.* all kinematic pairs are actuated and we know in advance their positions, velocities and so on. Closed kinematic chains, and in particular parallel robots, have passive joints constraining and cooperating to the desired end-effector behaviour. The parameters of all of these passive joints are not known in advance; quite the opposite, the passive joints parameters must be computed at each moment in function of the actuated joints parameters.

This step, computation of the passive joints parameters in function of the actuated joints, is the crucial part of the parallel robot kinematics. This complexity is derived from the closure of the chain that is not present in serial robots.

This chapter will be restricted to this problem and in how the hierarchical kinematic analysis can improve the solution of this problem. Another goal is to clarify the connection between the hierarchical canonical form and the kinematic structure of the parallel robot.

In the next section, the Kirchhoff-Davies laws will be presented to prepare the ground for the linear system at eq. (6.7) page 80. Kirchhoff-Davies laws lead to a system satisfying the requirements described in section 6.2.

In the sequence, section 6.8, we correlate the hierarchical kinematic analysis with the intrinsic kinematic structure of the parallel robot by using the classical concept of the Assur groups [Assur 1952].

6.5 Kirchhoff-Davies laws and parallel robots

The planar manipulator that will be described here is the 3RRR planar manipulator [Tsai 1999], which has $n = 8$ links and $e = 9$ kinematic pairs. This section will summarise the results of the application of the principles of the Kirchhoff-Davies current laws. These results are detailed in Appendix E and [Campos 2001]. The mechanism $3RRR$ in Figs. 20 and 21 has an associated coupling graph shown in Fig. 22.

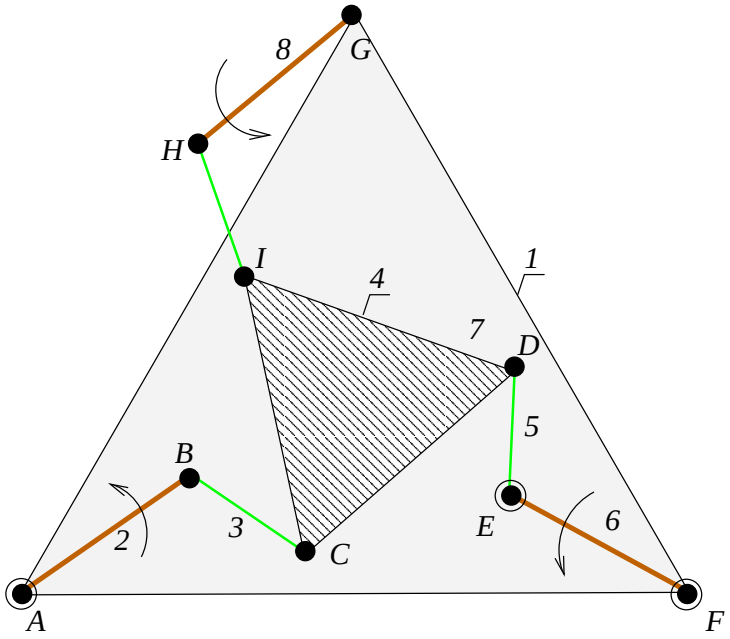

Figure 20: The planar Stewart platform $3RRR$ being actuated at joints A, F, G. The greater triangle 1 is attached to the base (fixed link or fixed platform) and the triangle 4 is the moving link or moving platform

The graph of the mechanism $3RRR$ has $l = 2$ circuits. We choose two arcs, one for each circuit, which are designated as the *chords* of the graph. The remaining arcs compose the selected tree of the graph, see details in [Seshu e Reed 1961] [Harary, Norman e Cartwright 1965] [Harary 1967] and appendix E. The chords chosen, for our example, are arcs A and G. Figure 23 shows chords A and G (with thick lines) and circuits M_A and M_G.

Thus, the matricial representation of the two circuits, M_A and M_G, of G_C is:

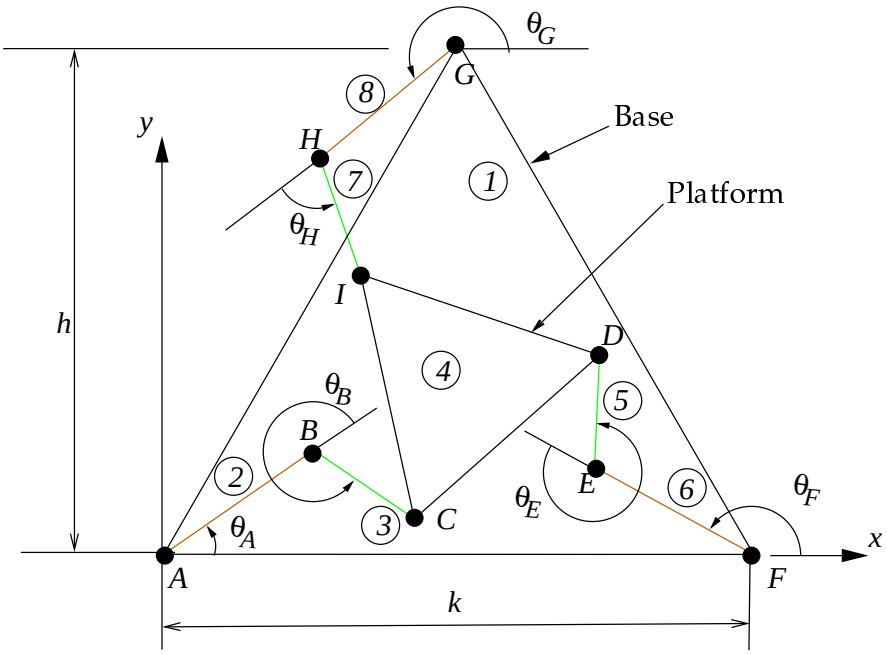

Figure 21: Details of the the dimensions (lengths and angles) as well as of the chosen coordinate system to represent the mechanism $3RRR$ in the xy-plane

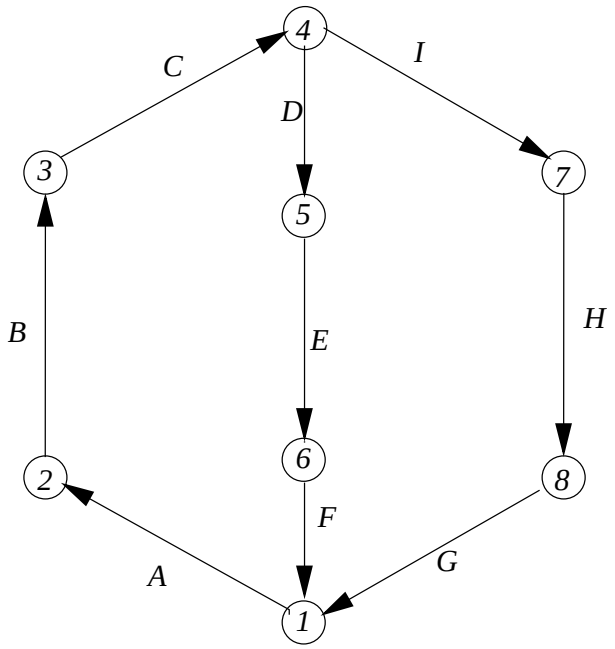

Figure 22: Coupling graph of mechanism $3RRR$.

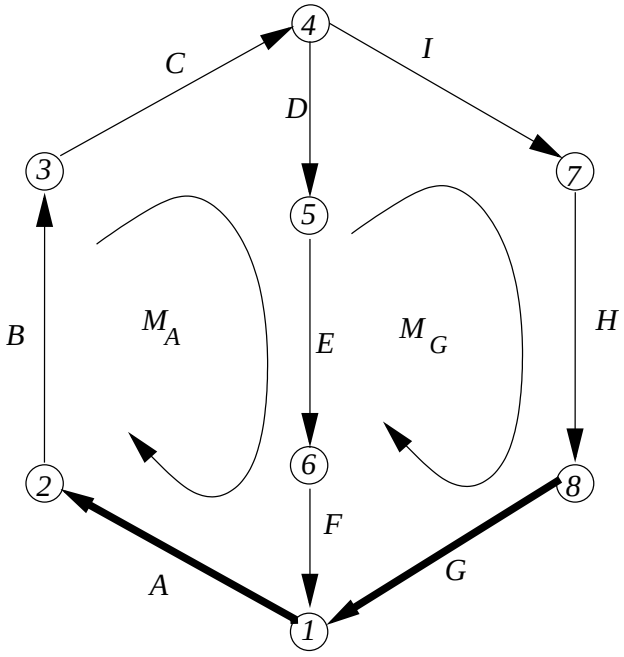

Figure 23: Chords and circuits of the movement graph G_M for the mechanism $3RRR$. This graph coincides with the coupling graph G_C as there is no kinematic pairs with $f > 1$.

$$B = \begin{array}{c} \begin{matrix} A & B & C & D & E & F & G & H & I \end{matrix} \\ \left[\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 \end{array}\right] \end{array} \begin{matrix} M_A \\ M_G \end{matrix} \tag{6.1}$$

The movement graph G_M for the mechanism $3RRR$ is also shown in Fig. 23; indeed, there is no substitution of arcs, as each one of the e arcs has $f = 1$ degree of freedom.

$$B_M = B = \begin{array}{c} \begin{matrix} A & B & C & D & E & F & G & H & I \end{matrix} \\ \left[\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 \end{array}\right] \end{array} \begin{matrix} M_A \\ M_G \end{matrix} \tag{6.2}$$

As the mechanism $3RRR$ is planar, $d = 3$, we consider only three Plücker's coordinates: $\mathcal{N}, \mathcal{P}$ and $\mathcal{Q}$ [Hunt 1978]. Therefore, the matrix of normalised screws, where the d rows represent the Plücker's coordinates and the F columns represent

the instantaneous screw at each kinematic pair of $f = 1$ degree of freedom, is given by

$$\hat{M}_D^T = \begin{array}{c} \begin{array}{ccc} \mathcal{P} & \mathcal{Q} & \mathcal{N} \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & 1 \\ ps_a & -pc_a & 1 \\ qs_{ba} + ps_a & -qc_{ba} - pc_a & 1 \\ qs_{fe} + ps_f & -qc_{fe} - pc_f - k & 1 \\ ps_f & -pc_f - k & 1 \\ 0 & -k & 1 \\ h & -\frac{k}{2} & 1 \\ ps_g + h & -pc_g - \frac{k}{2} & 1 \\ qs_{hg} + ps_g + h & -qc_{hg} - pc_g - \frac{k}{2} & 1 \end{array}\right] \end{array} \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{array} \tag{6.3}$$

where the matrix $\hat{M}_D$ is shown transposed in eq. (6.3) due to space restrictions. In eq. (6.3) $s_a = \sin\theta_A$, $s_b = \sin\theta_B$, $c_a = \cos\theta_A$, $s_{ba} = \sin(\theta_B + \theta_A)$, $c_{ba} = \cos(\theta_B + \theta_A)$, etc ... The dimensions $p = \overline{AB} = \overline{EF} = \overline{GH}$, and $q = \overline{BC} = \overline{FG} = \overline{HI}$ complete the dimensions shown in Fig. 21.

Matrix $\hat{M}_D$ represents the kinematics of the open kinematic chain *i.e.* a kinematic chain without closure of any kind. Up to this point, the matricial equation do not consider any interference between the joints. For short, matrix $\hat{M}_D$ cannot be used directly to obtain the kinematic equations of the (closed) kinematic chain. Some information about the closure of the chain has to be added, and this information is given by matrix B_M in eq. (6.2).

The Kirchhoff-Davies circuit law, which includes this closure information, is

$$[\hat{M}_N]_{(dl\times F)}[\Psi]_{(F\times 1)} = [0]_{(dl\times 1)} \tag{6.4}$$

where $[\Psi]_{(F\times 1)}$ is the vector of magnitudes of the twists *i.e.* angular and linear velocities. Appendix E, based on [Davies 2000], shows in some detail how to obtain $\hat{M}_N$ from B_M and $\hat{M}_D$.

Matrix $\hat{M}_N$ for the mechanism $3RRR$ is:

$$\hat{M}_N = \begin{bmatrix} 0 & p\,s_a & q\,s_{ba} + p\,s_a & q\,s_{fe} + p\,s_f \\ 0 & -p\,c_a & -q\,c_{ba} - p\,c_a & -q\,c_{fe} - p\,c_f - k \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -q\,s_{fe} - p\,s_f \\ 0 & 0 & 0 & q\,c_{fe} + p\,c_f + k \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdots$$

$$\cdots \begin{bmatrix} p\,s_f & 0 & 0 & 0 & 0 \\ -p\,c_f - k & -k & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ -p\,s_f & 0 & h & p\,s_g + h & q\,s_{hg} + p\,s_g + h \\ p\,c_f + k & k & -\frac{k}{2} & -p\,c_g - \frac{k}{2} & -q\,c_{hg} - p\,c_g - \frac{k}{2} \\ -1 & -1 & 1 & 1 & 1 \end{bmatrix} \tag{6.5}$$

and the vector of magnitudes is represented by the angular velocities at the kinematic pairs:

$$\Psi = \begin{bmatrix} \Psi_A & \Psi_B & \Psi_C & \Psi_D & \Psi_E & \Psi_F & \Psi_G & \Psi_H & \Psi_I \end{bmatrix}^T \tag{6.6}$$

where Ψ_i corresponds to the magnitude of the velocity at the kinematic pair i.

The linear system $[\hat{M}_N][\Psi] = [0]$ defines completely the (closed) kinematic problem. To solve it, one must select the joints that shall "carry on" the primary variables and those joints which will "carry on" the secondary variables. The primary variables are the joint velocities imposed by the actuators; the secondary variables are the velocities at the so-called passive joints.

6.6 Hierarchical kinematic analysis of parallel robots

Sections 6.5 described the Kirchhoff-Davies circuit law and its use to the kinematics of closed kinematic chains, with particular attention to parallel robots.

This section describes how to apply the hierarchical kinematic analysis to enhance the solution of parallel robot kinematics.

The Kirchhoff-Davies circuit law leads to a relationship between each joint of the kinematic chain, but it is not constrained to a specific choice of the actuators. Each choice of the actuators underlines a different mechanism in a same kinematic chain. Each mechanism has some particularity in the solution of the kinematic equations, which leads to a different hierarchical kinematic analysis.

In section 6.8, we briefly indicate the relationship among the several hierarchical canonical forms generated by all the possible divisions of the variables, and the Assur groups. A set of Assur groups is associated with each division. In this section, we explore the differences in the hierarchical kinematic analysis among the different mechanisms in a *same* kinematic chain. The kinematic chain, used as example here, is again the 3RRR planar Stewart platform.

Equation (6.4), with eqs. (6.5) and (6.6), describes a homogeneous linear system. The solution of this system, which is in fact the solution of the kinematics of the parallel robot, requires the choice, among the Ψ_i in eq. (6.6), of d primary variables *i.e.* those joints where the actuators are placed. The remaining (passive) joints carry the secondary variables (joint velocities) which are the solution of the system.

Equation (6.4) can be split in

$$[\hat{M}_N]_{sec}[\Psi]_{sec} = -[\hat{M}_N]_{pri}[\Psi]_{pri} \tag{6.7}$$

where $[\hat{M}_N]_{sec}$ is a $(dl \times dl)$ matrix and $[\hat{M}_N]_{pri}$ is a $(dl \times d)$ matrix composed by the columns of $[\hat{M}_N]$ corresponding to the primary joints. Vector Ψ is, accordingly, subdivided into $[\Psi]_{pri}$ and $[\Psi]_{sec}$.

In the 3RRR case, there is 6 rows and 9 columns in eq. (6.5) which can be divided into a square 6×6 matrix $[\hat{M}_N]_{sec}$ and a 6×3 matrix $[\hat{M}_N]_{pri}$. There is $\binom{9}{3} = 84$ possible sets of primary joints, *i.e.* 84 different 6×3 matrices $[\hat{M}_N]_{pri}$.

The subdivision of the matrix $[\hat{M}_N]$ into two parts has some similarity with

the serial manipulator system $\dot{x} = J\dot{q}$. In this case, the analogue of the serial manipulator Jacobian J is the square matrix $[\hat{M}_N]_{sec}$, the analogue of the vector $\dot{q}$ of serial robots is the vector $[\Psi]_{sec}$, and the analogue of the vector $\dot{x}$ of serial robots is the vector formed by $-[\hat{M}_N]_{pri}[\Psi]_{pri}$. This last vector may be considered the sum of the effects of the all the actuators on the mechanism.

We will describe 3 choices of the actuators (*i.e.* 3 different mechanisms of the same kinematic chain) and compare with their differences in the sequence.

6.7 Permutation of the system

The Algorithm 1 can rearrange the square matrix $[\hat{M}_N]_{sec}$ to a block-triangular form, unless it is undecomposable *i.e.* when the reachability matrix, eq. (2.6) at page 19, is full.

However, prior to rearrange the square matrix the algorithm generates two permutation matrices, P_r for rows and P_c for columns. While P_c permutation is restricted to the first member of eq. (6.7), P_r permutation affects both sides of eq. (6.7). The rearranged system is

$$P_r[\hat{M}_N]_{sec}P_cP_cT[\Psi]_{sec} = -P_r[\hat{M}_N]_{pri}[\Psi]_{pri} \tag{6.8}$$

which leads to

$$[\check{M}_N]_{sec}[\check{\Psi}]_{sec} = -P_r[\hat{M}_N]_{pri}[\Psi]_{pri} \tag{6.9}$$

So the right side of eq. (6.7) has to permute (P_r) its rows to maintain coherence with left side. As the dimensions of these matrices is not too large, in general, these permutations can be done by inspection. In our example, the first there rows of $[\hat{M}_N]_{pri}$ are switch to the last three rows in sections 6.7.2 and 6.7.3.

6.7.1 Actuators at joints A, F, G

Figure 24 shows the mechanism 3RRR with actuators at joints A, F, G.

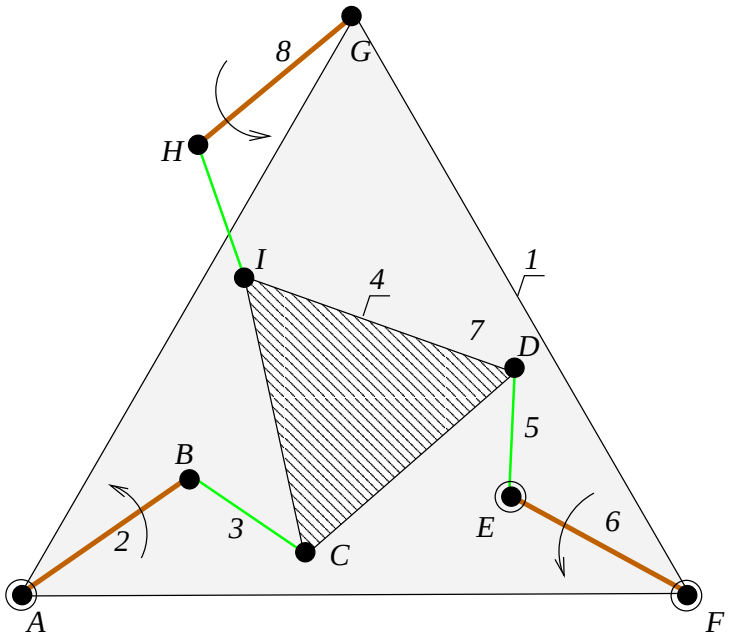

Figure 24: Mechanism $3RRR$ actuated at joints A, F, G

The secondary variables correspond to the columns 2, 3, 4, 5, 8 and 9 of eq. (6.5). The structure of the secondary matrix $[\hat{M}_N]_{sec}$ is

$$[\hat{M}_N]_{sec} = \begin{bmatrix} \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \tag{6.10}$$

Using algorithm 1 from chapter 2 at page 22, we notice that eq. (6.10) is undecomposable, as the reachability matrix, eq. (2.6) at page 19, is full. In other words, no independent row and column permutation can lead matrix (6.10) to a block-triangular form.

6.7.2 Actuators at joints D, E, F

Figure 25 shows the mechanism 3RRR with actuators at joints D, E, F.

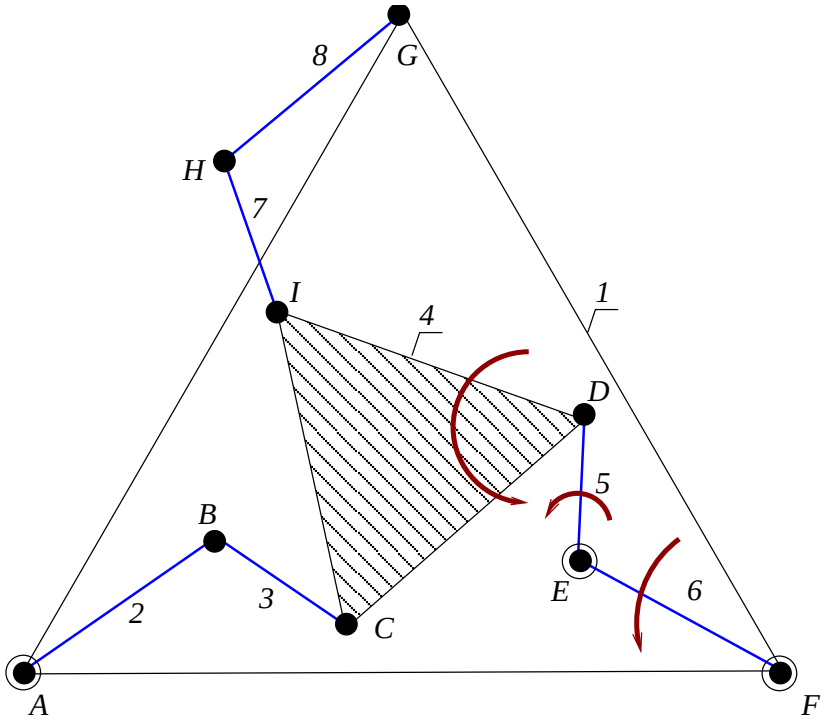

Figure 25: Mechanism $3RRR$ actuated at joints D, E, F

The secondary variables correspond to the columns 1, 2, 3, 7, 8, and 9 of eq. (6.5). The structure of the secondary matrix $[\hat{M}_N]_{sec}$ is

$$[\check{M}_N]_{sec} = \begin{bmatrix} 0 & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix} \tag{6.11}$$

In this case the Algorithm 1 from chapter 2 has no effect, as the matrix is already in its hierarchical canonical form. The matrix is a block-diagonal matrix

and the linear system becomes:

$$\left[\begin{array}{ccc|ccc} h & p\,s_g + h & q\,s_{hg} + p\,s_g + h & 0 & 0 & 0 \\ -\frac{k}{2} & -p\,c_g - \frac{k}{2} & -q\,c_{hg} - p\,c_g - \frac{k}{2} & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & p\,s_a & q\,s_{ba} + p\,s_a \\ 0 & 0 & 0 & 0 & -p\,c_a & -q\,c_{ba} - p\,c_a \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \left[\begin{array}{c} \Phi_G \\ \Phi_H \\ \Phi_I \\ \hline \Phi_A \\ \Phi_B \\ \Phi_C \end{array}\right]$$

$$= - \left[\begin{array}{ccc} -q\,s_{fe} - p\,s_f & -p\,s_f & 0 \\ q\,c_{fe} + p\,c_f + k & p\,c_f + k & k \\ -1 & -1 & -1 \\ \hline q\,s_{fe} + p\,s_f & p\,s_f & 0 \\ -q\,c_{fe} - p\,c_f - k & -p\,c_f - k & -k \\ 1 & 1 & 1 \end{array}\right] \left[\begin{array}{c} \Phi_D \\ \Phi_E \\ \Phi_F \end{array}\right] \tag{6.12}$$

Equation (6.12) shows two blocks of secondary variables (G, H, I) and (A, B, C). There is not a partial order between them, as the blocks corresponding to (A, B, C) and to (G, H, I) can be solved in any order. The system is completely decoupled. From the mechanical point of view the decoupling of the the system corresponds to the decoupling of the mechanism into actuators and *two* Assur groups as described below in section 6.8.

6.7.3 Actuators at joints *A*, *F*, *E*

Figure 26 shows the mechanism 3RRR with actuators at joints A, F, E. The structure of the secondary part of $[\hat{M}_N]$ is given by choosing the columns which correspond to the secondary joints B, C, D, G, H, I. These are the columns 2, 3,

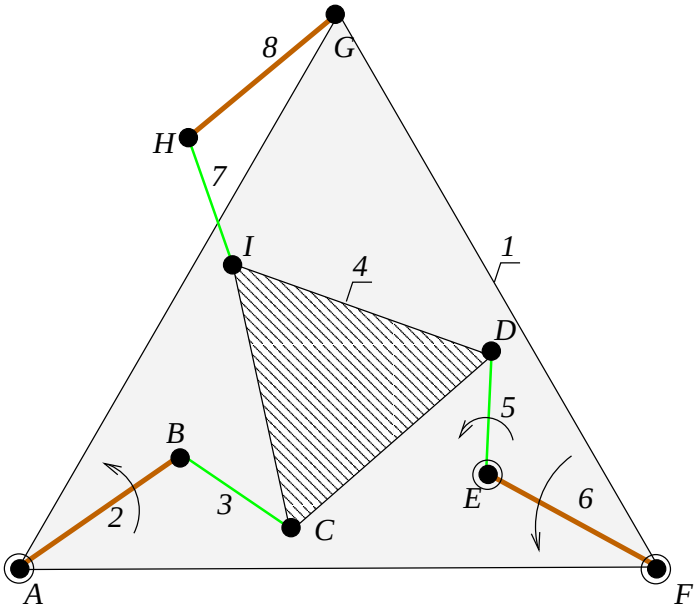

Figure 26: Mechanism $3RRR$ actuated at joints A, F, E

4, 7, 8, 9 in eq. (6.3). The structure of this square matrix is

$$[\check{M}_N]_{sec} = \begin{bmatrix} \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \tag{6.13}$$

Using Algorithm 1 from chapter 2 at page 22, system (6.4) rearranged be-

comes

$$\left[\begin{array}{ccc|ccc} h & p\,s_g + h & q\,s_{hg} + p\,s_g + h & 0 & 0 & -q\,s_{fe} - p\,s_f \\ -\frac{k}{2} & -p\,c_g - \frac{k}{2} & -q\,c_{hg} - p\,c_g - \frac{k}{2} & 0 & 0 & q\,c_{fe} + p\,c_f + k \\ 1 & 1 & 1 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & p\,s_a & q\,s_{ba} + p\,s_a & q\,s_{fe} + p\,s_f \\ 0 & 0 & 0 & -p\,c_a & -q\,c_{ba} - p\,c_a & -q\,c_{fe} - p\,c_f - k \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \left[\begin{array}{c} \Phi_G \\ \Phi_H \\ \Phi_I \\ \hline \Phi_B \\ \Phi_C \\ \Phi_D \end{array}\right]$$

$$= -\left[\begin{array}{ccc} 0 & -p\,s_f & 0 \\ 0 & p\,c_f + k & k \\ 0 & -1 & -1 \\ \hline 0 & p\,s_f & 0 \\ 0 & -p\,c_f - k & -k \\ 1 & 1 & 1 \end{array}\right] \left[\begin{array}{c} \Phi_A \\ \Phi_E \\ \Phi_F \end{array}\right] \tag{6.14}$$

Equation (6.14), similarly to eq. (6.12), shows two blocks of secondary variables: (G, H, I) and (B, C, D). However, in this case, there is a partial order between these blocks of variables, see section 2.2.2, as the block corresponding to (B, C, D) must be solved *before* the block corresponding to (G, H, I).

6.8 Assur groups

The kinematic analysis methods for closed chain mechanisms, more specifically to parallel robots, are often performed by particular methods. A discussion about the use of particular and/or general methods in Kinematics is found in [Fanguella e Galletti 1989].

Some developments had been made for chains with only one circuit or *loop* [Fanguella 1995] or for certain classes of planar manipulator with closed kinematic chains [Gosselin e Merlet 1994] [Gosselin 1996].

The different behaviour presented by the kinematic chain when different actuators are selected underline the intrinsic character of each mechanism derived. In fact, this diversity can be explained using Assur groups [Assur 1952] [Manolescu e Manafu 1963] [Baranov 1985] [Artobolevski 1988]

The technique of Assur groups [Manolescu 1968] [Hunt 1983] [Fanghella e Galletti 1990] [Innocenti 1995] tries to decouple the kinematic equations from complex kinematic chains into smaller modules whose behaviour can be more easily studied. This technique is quite widespread for planar kinematic chains, mainly the graphical approach [Baranov 1985]. However, computational approaches can be already found in [Zablonski, Belokonev e Tchekin 1989] [Martins e Macie 1996] [Martins e Macie 1996].

Manolescu (1963,1968) defines an Assur group as an open subset which can be added to a kinematic chain without affecting the mobility of the chain. The simplest planar Assur group is the binary dyad, see a pair of them in Fig. 27. More complicated Assur groups, both planar and spatial, can be found for instance in [Tischler 1995]. If the unconnected joints of an Assur group were attached directly to a base, a kinematic chain with zero degree of freedom would result.

Assur groups can be divided into minimal and non-minimal Assur groups. The latter ones may be divided into smaller (fewer number of bodies) Assur groups. The first ones are those which cannot be divided into minimal Assur groups. For example, Fig. 28 shows a non-minimal Assur group and highlights its decomposition into two minimal Assur groups, in this case two dyads.

Other division of planar Assur groups is relative to the order of these groups [Martins e Macie 1996] [Martins e Macie 1996]. For the planar Assur groups described in this work, Figures. 27 and 28 show first order Assur groups, while Fig. 29 show a second order Assur group. The solution of higher order Assur groups is, both graphically and computationally, more complex than lower order Assur groups [Martins e Macie 1995]. The concept of order is generalised to any spatial Assur group in [Tischler 1995]

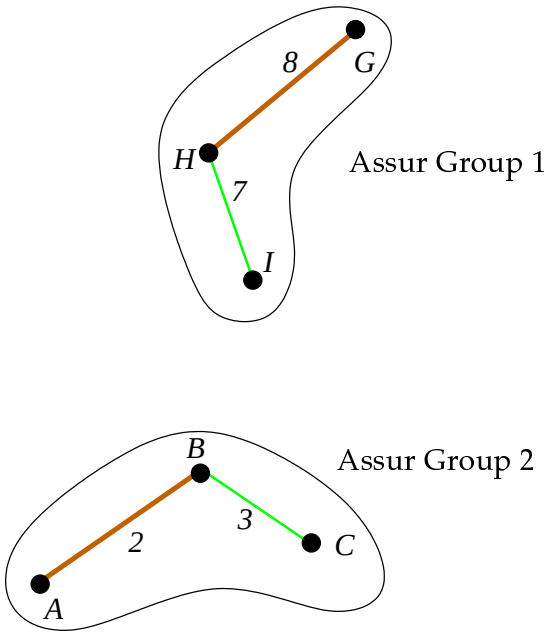

Figure 27: Pair of first order Assur groups derived from mechanism $3RRR$ actuated at joints D, E, F shown in Fig. 25. The pair of Assur groups, in this case two dyads, can be solved in any order, *e.g.* Assur group 1 can be solved after or before Assur group 2.

6.9 Conclusions about parallel robots

This chapter described the main characteristics of parallel robots and how the hierarchical kinematic analysis can be included in this context.

The kinematic model was based on the Kirchhoff-Davies circuit law. This modelling yields a linear system which can be improved by the hierarchical kinematic analysis.

The decomposition of this system has close relationship with the intrinsic structure of the mechanism. This structure-decomposition association is generated by the Assur groups which compose the mechanism.

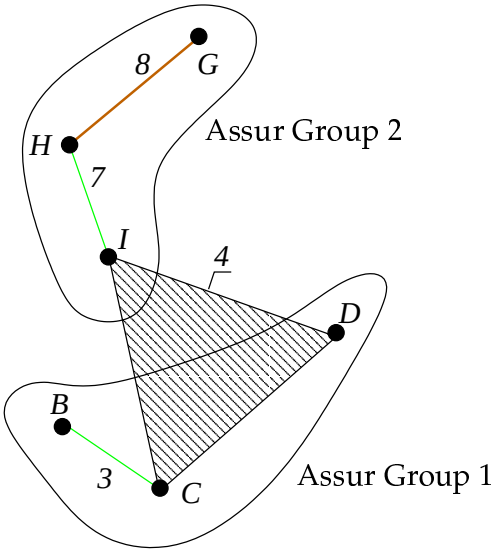

Figure 28: Pair of first order Assur groups derived from mechanism $3RRR$ actuated at joints A, F, E shown in Fig. 26. Notice that this Figure as a whole is non-minimal Assur group, but it can be split up into two minimal Assur groups, in this case two dyads which must be solved in order: Assur group 1 *always* before Assur group 2.

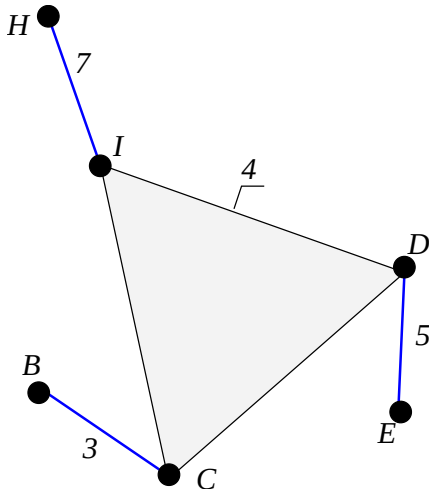

Figure 29: Second order Assur group derived from the mechanism $3RRR$ actuated at joints A, F, G shown in Fig. 24. Notice that this Figure as a whole is a minimal Assur group, so it cannot be split up into smaller Assur groups.

7 Conclusions

This thesis develops the hierarchical kinematic analysis for serial, redundant and parallel robots. The hierarchical kinematic analysis is a method based on digraphs which leads the Jacobian matrix to a better form, the hierarchical canonical form, to the analysis and resolution of the (inverse) kinematics of a wide class of manipulators.

Some of the characteristics of the hierarchical kinematic analysis are:

- the determinant is easier to obtain in the hierarchical canonical form.
- the hierarchy of diagonal blocks induces a hierarchy of singularities.
- the concept of *flow of variables* is developed.
- the variables are classified as free or affected variables according to each singularity. This classification has a direct correlation with the flow of variables.
- the pivoting process in a numerical LU decomposition [Golub e Loan 1983] of an upper block-triangular matrix is almost done without aggregating errors to the solution.
- the inversion of a 6×6 Jacobian matrix is, in general, substituted by simpler scalars or 2×2 diagonal submatrices inversions.
- In the neighbourhood of a singularity, not all Cartesian velocities need to be changed or adapted. Some may follow the path planning without

modifications, even when the online control system alerts the proximity of a singularity.

In chapters 5 and 6, the hierarchical kinematic analysis is extended to redundant and parallel robots. Both of these extensions required a review of the current methods of kinematic solution. For the redundant robots, we derived a variant of the extended Jacobian method to the redundancy resolution. For parallel robots, we had to adapt the Kirchhoff-Davies laws to solve for the passive joints parameters.

The method was applied to classical manipulators, such as the Puma robot and the 3RRR planar variant of the Stewart-Gough platform. In both cases we derive orignal correlations of the intrinsic geometrical structure of these manipulators. For the redundant robot analysis, we applied to a new robot architecture, the Roboturb manipulator for repairing eroded turbine blades [Guenther, Simas e Pieri 2000] [Martins e Guenther 2001], designed as a redundant robot for confined workspaces at the UFSC.

7.1 Perspectives and further work

Other topics are related to this thesis are worthy to mention. We present here a list of possible new topics and enhancements of the work of the thesis.

- Development of a methodology and/or some criteria to derive a screw coordinate system which can decouple the maximum number of input variables for any manipulator. For instance, given a Denavit-Hartenberg Jacobian, the application of the hierarchical kinematic analysis directly on this Jacobian may suggest an appropriate coordinate system to refine the analysis. In parallel some criteria to select the “best” coordinate system may be provided.

- Analysis of the intrinsic singularities of redundant robots serial chain to provide a strategy to singularity avoidance using the globally constrained Jacobian method.

- Formalisation of a singularity avoidance technique, expanding the method used in [Martins, Simas e Guenther 2001], to singularities which depend upon a conjugation of two or more joint variables as $x_{14} = gc_2 + hc_{23} = 0$ of the Puma robot, chapter 4.

- Expansion of the globally constrained Jacobian method to provide strategies for general manipulators in constrained or confined workspaces.

- Integration between the hierarchical kinematic analysis and the path planning.

- Integration between the hierarchical kinematic analysis and the design of control algorithms specific for a robot architecture.

- Application of similar results to the statics of mechanisms, including serial and parallel robots, following the duality between statics and kinematics and the Kirchhoff-Davies current laws [Davies 1983].

- Extension of the Algorithm 1 to the dynamic equations, as those listed in [Tischler et al. 2000].

- Extension of the above procedures to a specific manipulator comparing performance of the several choices of coordinate systems.

- Comparison among different manipulator configurations Jacobian matrices in the rearranged form.

- Generalisation of the combinatorial results obtained through a digraph approach to a more generic scope. For instance using bipartite graphs and matroids [Murota 2000] [Murota 1987]. A good possibility is the use of the combinatorial canonical form cited in [Murota e Scharbrodt 1998].

- Experimentation with more parallel robot architectures, specially spatial manipulators.

APPENDIX A – Screw Theory Basics

A.1 Kinematics and Statics of a rigid body

A review of some properties of the rigid body motion and static equilibrium must precede the description of screw theory. The properties presented in this section are the basis of the theory described in section A.2.

A body is considered rigid if it satisfies the rigidity condition *i.e* given any pair of distinct points $P(x_P, y_P, z_P)$ and $Q(x_Q, y_Q, z_Q)$ in the body the distance

$$d(P,Q) = \sqrt{(x_Q - x_P)^2 + (y_Q - y_P)^2 + (z_Q - z_P)^2}$$

is constant and independent of the time.

Kinematics and Statics of a rigid body have some similarities. Nonetheless it is rare to observe in undergraduate textbooks a closer relationship between Statics and Kinematics. However these similarities (more precisely dualities) are crucial for a deep understanding of both areas.

Some connections between Statics and Kinematics (of rigid bodies) appear at first glance. For instance the presence of a free vector and a line vector in both areas. Forces and *angular* velocities are examples of *line-vectors*. They have a line of action where they can slide freely without changing the static and/or kinematic state of the body. On the other hand, couples and *translational* velocities are examples of *free* vectors. Free vectors are those that do not depend upon a specific point of application. Section A.1.2 for Statics and section A.1.3

for Kinematics describe the application of these concepts. Section A.1.1 will give more detail about line vectors and free vectors.

Section A.1 introduces some concepts from Vectorial Mechanics. Most of the following texts include *ipsis litteris* extracts from Coe (1938), for instance Theorems 1 and 2.

A.1.1 Line vectors and free vectors

From High School Physics two vectors are considered equal if and only if they have same direction, same sense along this direction and same length. It is often adequate to combine other ideas in a same concept or be more specific when dealing with a physical concept.

In Mechanics it is convenient to restrict the definition above to a "free vector" as there are no constraints imposed on the "application point" of the physical quantity carried out by this vector. In other words a *free vector* depends only upon a direction, a sense and a length. Examples of free vectors are the translation of a rigid body, the force of gravity in a small region on the earth's surface and the velocity of light from a distant star (*e.g.* sunbeams).

A *line vector*, also called "sliding vector", is a combination of a straight line, a sense along this line and a length. Two line vectors will be equal only if they lie along the same axis, point in the same direction and have same length. Examples of line vectors are the angular velocity of a body at any instant as well as forces acting on a *rigid* body.

Another class of vectors is called attached vectors. An *attached vector* is a vector associated with a specific point of application. Examples include velocity profiles on a fluid flow, forces applied to a *deformable* body and the force of gravity throughout the solar system. Rigid body Statics and Kinematics has no examples of attached vectors so no further mention of attached vectors will be given in this work.

The operation of addition, subtraction, internal and external (dot and cross) multiplication of two vectors are performed on line and attached vectors exactly as if they were free vectors. The result of these operations, when a vector, is thought of as a free vector unless otherwise specified.

A.1.2 Statics

Any force vector field (including possibly some couples) can be substituted by a resultant force F and a resultant moment (or couple) M_P relative to a certain point. Due to the sliding characteristic of the force vectors the same resultants will be obtained if F is applied to any point along its line of action (Fig. 30). The classical result in Statics is

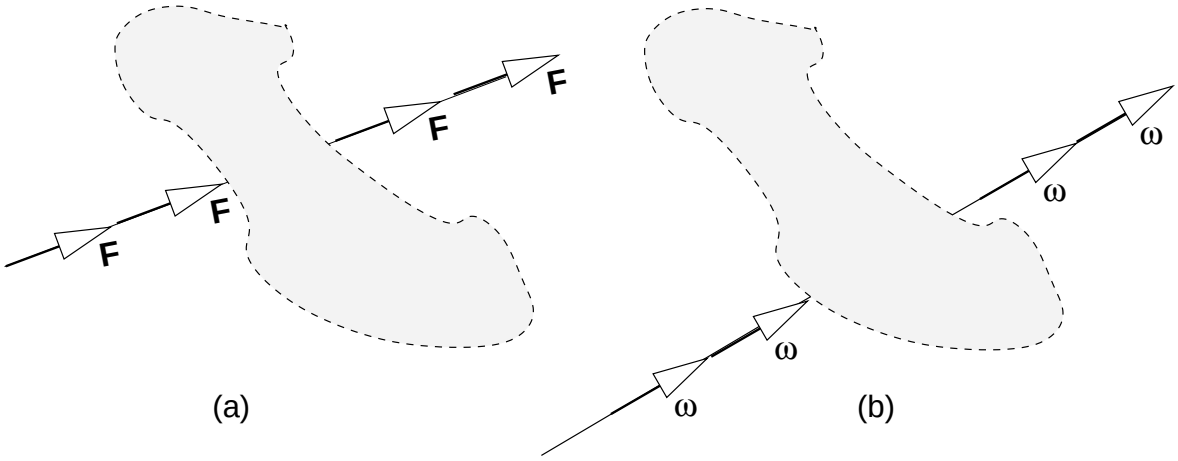

Figure 30: Line-vectors a) Forces b) Angular velocities.

Theorem 1 ([Poinsot 1806]) *Given a set of forces and pure couples applied to a rigid body it is always possible to find a line along which the resultant force and the resultant moment vectors will be directed.*

The line mentioned in Theorem 1 is called Poinsot's line. It is unique unless $F \equiv 0$ (pure couple)[1]. The Poinsot's line does not necessarily intersects any

[1]Using projective geometry it can be said that the pure couple is a force whose line of action is at infinite

material point of the body. It only depends upon the external forces and couples applied to the body. The shape of the body and even its convexity or continuity are immaterial for the computation of the Poinsot's line.

Couples can always be given as force binaries. The characteristics of a couple are the magnitude of the pair the forces, the orientation of the plane where the opposite forces actuate and their distance. The couple can be applied at any material point on the body without affecting the static equilibrium of the body. These facts characterize the couple as a *free vector*[2].

A.1.3 Kinematics

Any rigid body movement can be decoupled in translational and rotational parts. The velocity vector field in a rigid body can be substituted by a "resultant" translational velocity[3] τ and a "resultant" angular velocity ω relative to a certain coordinate system. Due to the line-characteristic of the angular velocity vectors this result will not be affected if ω is displaced along its line of action. The classical result in instantaneous Kinematics is

Theorem 2 ([Mozzi 1763]) *The velocities of the points of a rigid body at any instant are what they would be if the body were rotating about a certain fixed axis and simultaneously had a motion of translation along this axis.*

Chasles in 1830 expanded Theorem 2 to any *finite* displacement of a rigid body. Today Theorem 2 is often refereed as Chasles-Mozzi's Theorem or simply Chasles' Theorem. This line is unique unless $\omega \equiv 0$ (pure translation). Similarly to section A.1.2 this line do not necessarily intersects any material point of the body.

[2] A free vector is defined as a infinite pitch screw, section A.2

[3] We use τ instead of the normally accepted v for two reasons:

- to reinforce the *t*ranslational aspect of this velocity and
- to avoid confusion later when we represent a screw as a dual vector $\$ = (u, v)$

Translational velocity vectors are normally, in literature, associated to the centre of mass of the body. This division decouples translational and rotational kinetic energies and, in this sense, has some didactic appeal. The translational velocity represents the translational "state" of the *whole body* and not of a specific point. Any point pertaining to this body has the same translational velocity. The resultant velocity field differs from point to point in virtue of the rotational component of the velocity. The rotational component of the velocity is proportional to the perpendicular distance from the point to the Chasles' line. The translational velocity vector τ is another example of *free vector* as it is independent of the point of application.

A.2 Screw Theory

There is a correlation or similitude of procedures relating Theorems 1 and 2. This correlation was first discovered and formalised by Ball (1900) and remained untouched for decades until Dimentberg (1948) and Hunt (1978) recover this topic applying it to the theory of mechanisms.

The angular velocity ω and the linear velocity vector v_P of a point P on a moving body are three dimensional vectors. They can be assembled into a six-dimensional vector $\$^t$ called *twist*. The resultant force F and the couple C acting *at a point* on the body can be assembled into a similar six-dimensional vector $\$^w$ called *wrench*. The mathematics of these vector pairs as well as their physical and geometrical interpretation, also called dual vectors, is known as *screw theory*, see Ball (1900), Hunt (1990) and Phillips (1984,1990). A mathematical formalism known as a dual vector algebra is described in Dimentberg (1948), Woo & Freudenstein (1970) and Bottema & Roth (1979). An algebraic treatment of screw theory can also be found in Murray *et al.*(1994) or Selig (1996).

A.2.1 Twists × Wrenches

A.2.1.1 Twists

Kinematics is based on the concept of velocity. Velocity of a rigid body, however, has double meaning as the rigid body generally translates and rotates simultaneously.

Thus angular velocity ω and translational velocity τ of the body must be differentiated. Lemma 10, section B, shows how to uniquely represent τ and ω via a dual vector $\$^t = (\omega; \mathbf{v}_P)$, composed by the angular velocity ω of *the whole body* and the linear velocity $\mathbf{v}_P$ of *a point* P attached to the body. Both velocities, ω and $\mathbf{v}_P$, need to be represented in the same coordinate system. On the Chasles' line, and only there, ω and $\mathbf{v}_P$ will be parallel. Moreover $\mathbf{v}_P$ will have the same value τ on any point along the Chasles' line.

Definition 3 *A* twist *is a dual vector* $\$^t = (\omega; \mathbf{v}_P)$ *composed by the angular velocity* ω *of the body and the linear velocity* $\mathbf{v}_P$ *of a point* P *attached to the body, both vectors represented in the same coordinate system.*

The six-vector $\$^t = (\omega; \mathbf{v}_P) = (\omega; \mathbf{v}_Q + r_{QP} \times \omega)$ relative to a generic point P is called the twist of the motion. The displacement vector $r_{QP} = P - Q$ relates the point where the linear velocity $\mathbf{v}_Q$ is *a priori* known (Q) and the desired point (P)[4]. The twist is normally represented in ray order, see section A.2.5.

The second part (after ";") represent the velocity of a point in the body instantaneously coincident with the origin of the frame [Hunt 1978]. This particular choice has some advantages, specially in the definition of the screw coordinate transformation (section A.3). The symbol ; just separates two vectors that have different physical dimensions. For instance ω can have dimension rad/s and v_P dimension m/s.

[4]Neither P nor Q have to be on the Chasles' line.

A.2.1.2 Wrenches

Statics of a rigid body is associated with (external) loads applied to this body. The term load, as the term velocity discussed in section A.2.1.1, has also a double meaning. A load can be a force or a couple (binary, moment of a force).

It is a common practice in elementary Statics to select an adequate point and perform, relatively to this point, the summation of forces and summation of moments. However, differently from the twist case, the wrench free vector (resultant couple C) varies from point to point and the wrench line vector (resultant force F) is independent of the chosen point. On the Poinsot's line (and only there, see section B) C and F are parallel.

Definition 4 *A* wrench *is a dual vector,* $\$^w = (F; M_P)$*, composed by the resultant (external) force* F *applied to a body and the resultant couple* M_P *computed at a point* P *of the coordinate system where both vectors are represented.*

The vector pair $\$^w = (F; M_P) = (F; M_Q + r_{QP} \times F)$ is the dual vector associated with the static state of the system at P starting from an *a priori* known summation of couples at a different point Q. This order of the vectors is called *ray order*, see section A.2.5.

A wrench is represented in relation to the origin of the coordinate system unless otherwise stated. This is also not obligatory but it is convenient [Hunt 1978].

A.2.2 Screws

A twist and a wrench are represented by a purely geometrical object called *screw* plus a (dimensional) magnitude.

Definition 5 *A* screw $\$$ *is a geometrical entity composed by a line and a number* h *called* pitch *which has length dimension.*

The relationship between a screw and a twist or between a screw and a wrench is *grosso modo* similar to the notion of a vector as an entity having magnitude and direction and its representation of the velocity or force acting on a body in space. Any physical quantity that requires an line of action and a pitch can represented by a screw. The *same screw* can be used indistinctly to represent a twist or a wrench applied to a body. The dimension of the magnitude (for instance rad/s and N) makes the distinction.

The notion of pitch h is associated with the relationship between both quantities along the screw axis. In kinematics $h = \tau/\omega$. In statics $h = C/F$. In both cases these quotients only make sense when both vectors are parallel *i.e.* along the Chasles' line (where $v_P = \tau$) or Poinsot's line (where $M_P = C$), respectively. See Fig. 31.

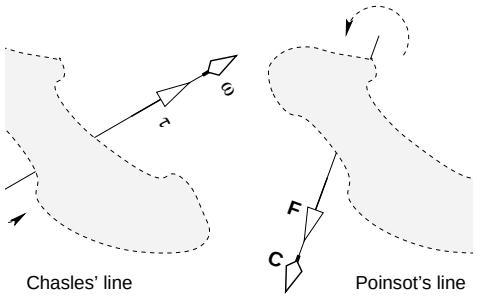

Figure 31: Chasles' line and Poinsot's line.

We now state an important Lemma

Lemma 6 *Any screw can be represented by a dual vector* $\$ = (u; v)$ *where the*

pitch can be calculated by

$$h = \frac{u \cdot v}{u \cdot u} \tag{A.1}$$

and the minimum length vector $r_\perp$ *that points from the origin of the coordinates to the axis of* \$ *is given by*

$$r_\perp = \frac{u \times v}{u \cdot u} \tag{A.2}$$

Proof:

Only the kinematic case will considered ($u = \omega$ and $v = \tau$). The same reasoning could be applied to the static case using $u = F$ and $v = C$ instead. The development for the static case is left as an exercise to the interested reader. Section B gives a more generic approach.

The linear velocity v_P of a point P in a rigid body can be decomposed into two parts: the translational velocity of the body (τ) and the rotational velocity component ($v_\perp$). The rotational velocity component $v_\perp$ is induced purely by the angular velocity ω.

Due to the definition of the pitch $\tau = h\omega$. Let r be a position vector starting from any point along the Chasles' line and pointing to P. The rotational velocity component will be $r \times \omega$.

$$v = h\omega + r \times \omega \tag{A.3}$$

The internal product with ω gives

$$\omega \cdot v = \omega \cdot (h\omega) + \omega \cdot (r \times \omega)$$

As the second term $\omega \cdot (r \times \omega)$ is null then

$$h = \frac{\omega \cdot v}{\omega \cdot \omega} \tag{A.4}$$

and the first part (pitch) of the lemma is demonstrated.

Applying the *external* product to eq. (A.3)

$$\omega \times v = \omega \times (h\omega) + \omega \times (r \times \omega) \tag{A.5}$$

This time the first term $\omega \times (h\omega)$ is null.

The vectorial identity

$$a \times (b \times c) = (c \cdot a)b - (a \cdot b)c$$

can be applied to the second term in eq. (A.5).

In this case

$$\omega \times (r \times \omega) = (\omega \cdot \omega)r - (\omega \cdot r)\omega$$

By definition $r_\perp$ will occur when $\omega \cdot r = 0$. So

$$r_\perp = \frac{\omega \times (r \times \omega)}{\omega \cdot \omega} = \frac{\omega \times v}{\omega \cdot \omega}$$

Finally the screw axis can be parameterised as

$$\$^t = -r_\perp + \alpha\tau$$

where α is the free parameter and the minus sign means that this time we are taking the direction *from the point* P to the screw axis and not vice-versa.

This line (screw axis) is unique as $r_\perp$, given by eq. (A.2), and τ are unique.

QED

A.2.3 Plücker coordinates

The geometry of rigid body movements is the geometry of the screws as above defined. This geometry is basically a line geometry with some dimensional remarks relative to the pitch and magnitude.

A.2.3.1 Plücker line coordinates

The representation of lines in Euclidean space by dual vectors in $\mathbb{R}^3$ is due to Plücker (1865). Other representations exist such as those based on quaternions or Lie (or Grassmann) algebras. Plücker line coordinates were chosen due to its natural extension to the screw representation.

A line can be expressed in Plücker coordinates as

$$\$^l = (u; v) = (\mathcal{L}, \mathcal{M}, \mathcal{N}; \mathcal{P}, \mathcal{Q}, \mathcal{R}) \tag{A.6}$$

where u represents the direction vector of the line and $v = r \times u$ is the "moment" of the line in respect to the origin of the frame.

Observe that by definition the pair of vectors making up the Plücker coordinates *of a line* are *always* orthogonal. In other words u and v satisfy $u \cdot v = 0$.

In terms of coordinates, $u = (\mathcal{L}, \mathcal{M}, \mathcal{N})$ and $v = (\mathcal{P}, \mathcal{Q}, \mathcal{R})$, the constraint above becomes

$$\mathcal{LP} + \mathcal{MQ} + \mathcal{NR} = 0 \tag{A.7}$$

Eq. (A.7) is known as the *quadratic identity*. It is a quadratic constraint in terms of the components of the screw.

The representation of lines using eq. (A.6) is homogeneous and can be normalised (see section A.2.4).

A.2.3.2 Plücker screw coordinates

The screw \$ is a geometrical entity with a pitch and an axis. Screw pitch is described on eq. (A.1) in terms of the Plücker coordinates of the screw axis Plücker coordinates of the screw axis is given in (A.6). The two equations can be melted together in a unique representation of the screw in terms of Plücker coordinates. These Plücker coordinates are designated here as Plücker screw coordinates.

Plücker screw coordinates are not the only way of representating screws. Ball (1900) used a set of six base screws These screws have non zero pitch and are directed along the coordinate axes.

Let v^* be the point dependent part of the dual vector (screw), that is, linear velocity in Kinematics or couple in Statics. In this section the superscript $(^*)$ represents a screw coordinate, as opposite to the Plücker line coordinates of its axis (u, v). The line vector part is u, angular velocity in Kinematics or force in Statics. The line vector part (u) can be decomposed into two orthogonal components: one parallel to the point independent part (u) and another component (v) perpendicular to (u). Plücker screw coordinates are (u^*, v^*). The parallel component, by equation (A.1), is obligatorily hu. The line vector $u = u^*$ as the direction cosines of the screw or of its line (axis) are the same so there is no need to keep the asterisks.

Symbolically the free vector part is

$$v^* = v + hu$$

and then the screw representation as a dual vector (in ray-coordinates, see

section A.2.5) becomes

$$\$ = (u; v^*) = (u; v + hu) \tag{A.8}$$

In terms of the Plücker coordinates explicitly

$$\$ = (\overbrace{\mathcal{L}, \mathcal{M}, \mathcal{N}}^{u}; \overbrace{\mathcal{P}^*, \mathcal{Q}^*, \mathcal{R}^*}^{v^*}) = (\mathcal{L}, \mathcal{M}, \mathcal{N}; \mathcal{P}-h\mathcal{L}, \mathcal{Q}-h\mathcal{M}, \mathcal{R}-h\mathcal{N}) \tag{A.9}$$

Here terms $\mathcal{P}^*, \mathcal{Q}^*, \mathcal{R}^*$ represent the coordinates of the point dependent part v^* of the screw. The vector $v = \{\mathcal{P}, \mathcal{Q}, \mathcal{R}\}$ describes the "moment" of the line versor relative to the origin of the coordinate frame as in section A.2.3.

Comparing eqs. (A.6) and (A.8) it can be deduced that Plücker coordinates of a line $\$^l$ form a particular subset of screws. Lines $\$^l = (u; v)$ can be considered as screws of null pitch. Lines interrelationships (like linear dependence, for instance) represent a particular case of screws interrelationships. This also justifies the use of the same symbol ($\$) to describe twists ($\$^t$), wrenches ($\$^w$) or lines ($\$^l$) (although these three quantities have different dimensions).

So Plücker coordinates may represent both lines and screws. The number of independent variables however differ. A line has 4 independent parameters. The screw has 5 independent parameters. The extra value corresponds to the pitch h. Plücker coordinates are composed by 6 elements so they are dependent one another. Both representations have less than 6 independent parameters. Some constraints among these parameters must exist. The number of constraints is $6 - 4 = 2$ in Plücker line coordinates and is $6 - 5 = 1$ in Plücker screw coordinates. The common constraint is the normalisation (section A.2.4). For Plücker line coordinates there is the perpendicularity between u and v ($u \cdot v = 0$). Normalised screws are discussed again in section A.2.4.

A.2.4 Normalised screws

Normalisation is the only constraint imposed on the Plücker coordinates representation of screws. Lines require an extra constraint that is the orthogonality between u and v or $u \cdot v \equiv 0$ or, equivalently, $h = 0$.

Normalisation is a consequence of the homogeneity of the Plücker coordinates ie

$$\$ \equiv \rho\$ \qquad (u; v) \equiv (\rho u; \rho v) \qquad \forall \rho \in \mathbb{R} \tag{A.10}$$

Eq. (A.10) indicates that the coordinates of a screw are *homogeneous*. Mathematically this means that the linear space spanned by screws is a projective space. Gibson and Hunt (1990) proved that this space is $\mathcal{P}^5$, the projective real space of dimension 5 (one dimension for each parameter that describe the screw).

Substituting $\omega \rightarrow u; v \rightarrow v^*$ in eq. (A.3) an equivalent equation can be found.

$$v = hu + r \times u$$

If instead of u we have a different vector $u' = \rho u$ where ρ is a real constant, the moment $v^{*'}$ of this (enlarged or reduced) new vector u' is

$$v^{*'} = h\rho u + r \times \rho u = \rho v^*$$

In both cases, (u, v^*) or $(u', v^{*'})$ $[= (\rho u, \rho v^*)]$, the *same* line is being described. The norm of the vector u is immaterial. Vector u is used only to represent the direction cosines of the straight line.

Let ρ be the *magnitude* of the screw defined by

$$\rho \triangleq \begin{cases} \|u\| & \text{if} \quad \|u\| \neq 0 \\ \|v\| & \text{if} \quad \|u\| = 0 \end{cases} \tag{A.11}$$

Definition 7 *A screw with its magnitude ρ factored out, following eq. (A.11), is as a* normalised screw.

By the definition given in eq. (A.11) a screw with null magnitude ($\rho = 0$) does not make sense or does not exist.

Eq. (A.11), in terms of the pitch h, can be derived from eq. (A.1)

$$\rho \triangleq \begin{cases} \|u\| & \text{if } h \text{ is finite} \\ \|v\| & \text{if } h \text{ is infinite} \end{cases} \tag{A.12}$$

All infinite pitch ($h = \infty$) screws represent something that does not depend upon an application point. Let us exemplify this fact with a screw, in ray coordinates, representing a wrench in metric units of the form

$$\begin{array}{rccccccc} \$^w = & (0, & 0, & 1 & ; & 0, & 0, & a) \\ & F_x & F_y & F_z & & M_x & M_y & M_z \end{array} \tag{A.13}$$

Eq. (A.1) indicates that $h = a$. Consider a wrench in eq. (A.13) being carried by a screw with a relatively high pitch. For instance $h = a = 10^{10}m$.

One question appears: *What does it mean a wrench whose screw has a so large pitch?* The formal answer is: the summation of forces will be a resultant force along the z-axis with unit intensity ($F_z = 1N$) and the resultant couple will be also along the z-axis but with greater intensity ($M_z = 10^{10}N.m$). In other words the resultant force can be considered negligible compared to the resultant couple.

A second question can be posed: *In which conditions such a disparity of values between resultant couple and resultant force can be obtained?* Answer: A couple depend upon the intensity of forces and the application point. As the intensity of the forces is comparatively small the line of action of this resultant force is far from the point where the couples are being computed. As discussed

in the end of section A.2.1.2 this point generally is the origin of the coordinate system.

A fast computation indicates that the minimal (ie perpendicular) distance from the origin of the coordinate system to the Poinsot's line is $M_z/F_z = 10^{10}N.m/1N = 10^{10}m$!! Our rigid body is possibly restricted to a small region around the origin of the coordinate system and then far from the Poinsot's line. The computation of resultant forces and couples in any point pertaining to the rigid body will make little difference in the result $\$^w = (0, 0, 1N; 0, 0, 10^{10}N.m)$

Let us now take a different approach. The wrench $\$^w = (0, 0, 1; 0, 0, a)$ can be divided by the norm of v ie $\$^w = 10^{10}N.m \cdot (0, 0, 1/a; 0, 0, 1)$. In the example $1/a = 1/10^{10}m^{-1}$ is insignificant compared to unity in the last coordinate. The tendency $a \to \infty$, or more precisely $h \to \infty$, means that the free vector part is more significant than the line vector part.

In the extremal case ($h = \infty$) there is no presence at all of the line vector part. This justifies the second condition in the definition of the magnitude of a vector, eq. (A.11).

Screw components have not the same physical dimension. The dimensions of vectors u and v are different one another. Dimensional discrepancy remains even when the screw is normalised. In this case u normalised is non dimensional and v becomes a length dimensional vector.

A screw can be a representation of a physical quantity like a twist or a wrench. In this case ρ has the physical dimension of this quantity. We use the operator $[\cdot]$ to designate the physical dimension associated to the variable inside the square brackets.

By the definition of the magnitude of the screw given by eq. (A.12) ρ has the dimension of the line vector part of the screw if it is a finite pitch screw and the dimension of the free vector part of the screw if it has infinite pitch. For instance if the screw is carrying a twist then $[\rho] = rad/s$; if it is is carrying a wrench $[\rho] = N$ (newtons). However if the twist is a pure translation $[\rho] = m/s$ or if the

wrench is a pure couple $[\rho] = N \cdot m$

A.2.5 Ray order × Axis order

As discussed in section A.2.4 twists and wrenches behave quite similarly. Both can be represented by a screw meaning that there is an axis and a (dimensional) pitch. Both are composed by a free vector part and a line-vector part. Table 2 summaries some similarities and differences between them.

	Kinematics	Statics
quantity	twist	wrench
line-vector	ω	F
free vector	τ	C
point dependent variable	v_P	M_P
magnitude dimension	rad/s	N
pitch dimension	m	m
basic theorem	Chasles'	Poinsot's
screw (ray order)	$\$^t = (\omega; v_P)$	$\$^w = (F; M_P)$
screw (axis order)	$\$^t = (v_P; \omega)$	$\$^w = (M_P; F)$

Table 2: Comparison between Statics and Kinematics. All variables are represented in SI units and are taken relative to a generic point P

One characteristic of the dual vector representation relates to point dependency. Generally all elements of the dual vector associated to a screw change after a coordinate transformation. This is not particularly true if the coordinate transformation is a pure translation of axes. In this case one of the two vectors is not affected by the transformation. In Kinematics the translational velocity τ will be the same as all points of the rigid body possess the same translational velocity (see Fig. 32. Similarly, in Statics,the summation of forces will be the same no matter what point is chosen to compute it (see Fig. 33. In the representation of twists, $\$^t = (\omega; \mathbf{v}_P)$ or $\$^t = (\mathbf{v}_P; \omega)$, or wrenches, $\$^w = (F; M_P)$ or $\$^w = (M_P; F)$, the part corresponding to ω or F is called the *point independent* part of the dual vector.

In other words, couples and translational velocities do not suffer any change

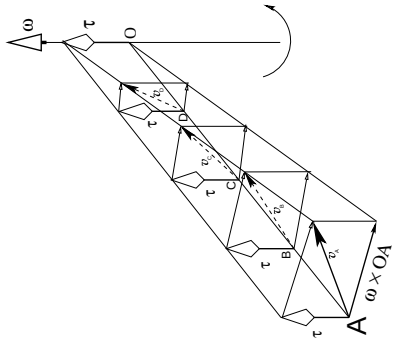

Figure 32: Twist represented by a screw (plus a dimensional magnitude). The velocity of a point A is given by $v_A = \tau + \omega \times \vec{OA}$

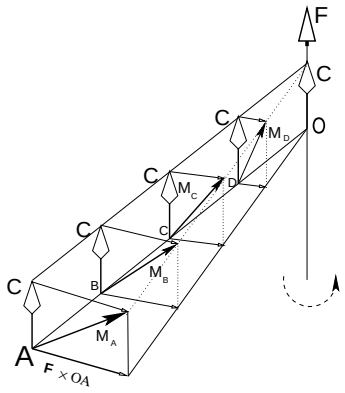

Figure 33: Wrench represented by a screw (plus a dimensional magnitude). The summation of couples at a point A is given by $M_A = C + F \times \vec{OA}$

in their components because they are an intrinsic characteristic of the *whole body* and not of a specific choice of the origin of the coordinate system.

We can define two orders to represent a screw using dual vectors

Definition 8 *A screw* $\$ = (u, v)$*, represented by its coordinates, is in* ray order *when the* second *part (v) is a free vector. In this case it can be said that the screw has a* representation in ray-coordinates.

Definition 9 *A screw* $\$ = (u, v)$*, represented by its coordinates, is in* axis order *when the* first *part (u) is a free vector. In this case it can be said that the screw has a* representation in axis-coordinates.

The choice between ray and axis order is fairly arbitrary, but need to be consistent. Wrenches can be described (as row vectors) in axis order and twists (as columns vectors) in ray order. The argument for this is based on the reciprocal product (or mutual moment), eq. (A.15), falling out in the matrix algebra. However, unless otherwise specified ray order will be adopted ie $\$^t = (\omega; \mathbf{v}_P)$ for twists and $\$^w = (F; M_P)$ for wrenches.

Observe that even a dual choice between ray and axis-order is not obligatory. A representation that mixes free vector and line-vector coordinates, like $\$ = (u_1, v_1, u_2, v_2, u_3, v_3)$ or any other permutation, is equally valid. As pointed out in [Gibson e Hunt 1990] the screw can be represented as a point in the projective real space of dimension 5 ($\mathcal{P}^5$). This reinforces the arbitrariness of the coordinates order. Furthermore the "finiteness" of some coordinates (or coordinate axes) is also immaterial. They can be always described in a projective space. See [Semple e Kneebone 1952] for more details in projective geometry.

An important warning: *the choice of representation must be* coherent *through the text.* Otherwise this will imply in a series of unnecessary algebraic confusions or theoretical misunderstandings. Duffy (1990) discusses an example of this kind of misunderstanding in current literature.

A.2.6 Virtual Work

Let $p \triangleq \overrightarrow{OP}$ and $q \triangleq \overrightarrow{OQ}$ be the vectors between the origin O of the coordinate system and points P and Q. Let $\$^w = (F; M_P) = (F; p \times F + C)$ be a wrench composed by a couple C computed at the origin of the coordinate system and a (resultant) force F whose line of action passes through a point P. Similarly $\$^t = (\omega, v_Q) = (\omega; q \times \omega + \tau)$ is a twist composed by the translational (linear) velocity τ of the whole body and an angular velocity ω whose line of action passes through a point Q. Both screws are represented in the same coordinate system with origin at O.

The work done by the wrench $\$^w$ as it moves through the twist $\$^t$ over a virtual time period δt (Figs. 34 and 35is given by

$$\mathcal{W} = F \cdot v_Q + M_P \cdot \omega \tag{A.14}$$

In matricial form (using ray coordinates and treating both dual vectors as column vectors)

$$\mathcal{W} = \$^{wT} \begin{bmatrix} 0_3 & I_3 \\ I_3 & O_3 \end{bmatrix} \$^t \tag{A.15}$$

Here I_3 and 0_3 are respectively the identity and null square matrices of order 3.

Expanding eq. (A.14) or (A.15) (see also Fig. 36)

$$\begin{aligned} \mathcal{W} &= (F \cdot (q \times \omega + \tau) + (p \times F + C) \cdot \omega) && \text{(A.16)} \\ &= (F \cdot \tau + C \cdot \omega + (q - p) \cdot (\omega \times F)) && \text{(A.17)} \end{aligned}$$

When wrenches are described as row vectors in axis order and twists as

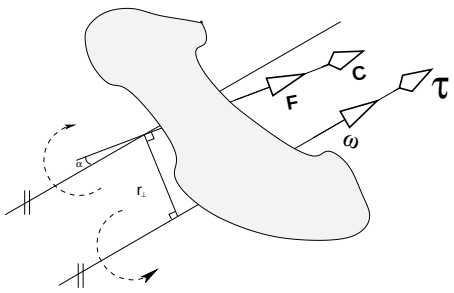

Figure 34: A wrench $\$^w$ acting on a twist $\$^t$.

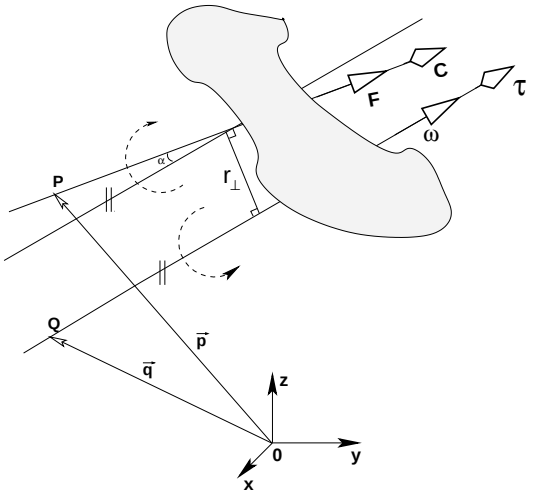

Figure 35: Virtual work of a wrench acting on a twist.

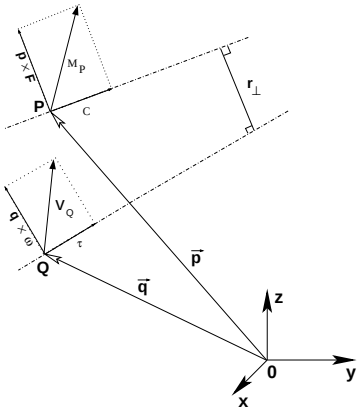

Figure 36: Details of the point dependent vector decomposition relative to the wrench and to the twist.

columns vectors in ray order the eq. (A.15) becomes

$$\mathcal{W} = \$^{w}{}_{axis}\$^{t}{}_{ray} \tag{A.18}$$

Eq. (A.16) can be interpreted making explicit all combinations of factors that generate the virtual work in geometrical terms. Quantities are considered along their respective screw lines. Forces (F) and couples (C) are computed at any point along the Poinsot's line and angular velocities ω and translational velocities τ are computed at any point along the Chasles' line. Remember that translational velocity τ coincides with the linear velocity v of any point along the Chasles' line.

In other words we have two lines: Poinsot's and Chasles'. At Poinsot's line a point is selected and relative to this point we have two parallel vectors (F and C) acting on the rigid body. On the other side we have a different point selected at the Chasles' line with two parallel vectors (ω and τ) representing completely the rigid body movement. The wrench $\$^{w} = (F; C)$ acts on a body that instantaneously twists around following $\$^{t} = (\omega; \tau)$. The remaining question is: *What does it mean geometrically the virtual work generated by the wrench* $\$^{w}$ *acting on the twist* $\$^{t}$

F and ω		$=$	$-F\omega r_{\perp} \sin\alpha$
F and τ	$= -F\tau \cos\alpha$	$=$	$F\omega h_t cos\alpha$
C and ω	$= C\omega \cos\alpha$	$=$	$F\omega h_w cos\alpha$
C and τ		$=$	$\mathbf{0}$

Table 3: Combination of factors to generate the virtual work

Table 3 lists the virtual work generated by each variable combination. There $r_{\perp}$ is the perpendicular distance between Poinsot's and Chasles' lines (screw axes), α is the torsion angle between both axes, h_t is the pitch of the twist and h_w is the pitch of the wrench (see Fig. 34). It is worth noticing that both point independent quantities cannot do any work each other. A couple C can only impose a rotational "tendency" to the movement of the rigid body and cannot

exert any effect to the translational state of the body ie cannot induce any change to the translational velocity τ.

Combining all factors in Table 3 in a same expression the rate of the work done $\mathcal{W}$ by a wrench onto a twist is

$$\mathcal{W} \triangleq F\omega[(h_t + h_w)\cos\alpha - a\sin\alpha] = F\omega\varpi \tag{A.19}$$

The instantaneous quantity ϖ is called the virtual moment by Ball (1900). It represents the geometrical information related to a wrench and a twist.

If the virtual work of a wrench acting on a twist is zero both screws are said to be reciprocal, and the wrench does no work on the body.

In a condensed way, the rate of the work done $\mathcal{W}$, eq. (A.19), will be zero in three cases

- $\rho(\$^t) = 0$
- $\rho(\$^w) = 0$
- $\varpi = 0$

The first two cases represent trivially a null magnitude of the twist or the wrench. In other words one of the quantities (wrench or twist) simply does not exist. The third case describes the geometrically relative locations of the screws where both are reciprocal.

A.3 Screw Coordinate Transformation

The coordinates of a screw depend upon the chosen coordinate systems. It is then necessary to know *how* to transform screws from one coordinate system to another. Tsai (1999) uses a different approach to derive the screw coordinate transformation based purely on vector relationships and on the D'Olindes-Rodrigues parameters.

A.3.1 Ray-axis screw coordinate transformation

Let us consider first the transformation between ray and axis-coordinates (section A.2.5) with dual vectors represented as column vectors.

$$\begin{bmatrix} \omega \\ v_P \end{bmatrix}_{\text{ray}} = \begin{bmatrix} 0_3 & I_3 \\ I_3 & 0_3 \end{bmatrix} \begin{bmatrix} v_P \\ \omega \end{bmatrix}_{\text{axis}} \tag{A.20}$$

$$\begin{bmatrix} M_P \\ F \end{bmatrix}_{\text{axis}} = \begin{bmatrix} 0_3 & I_3 \\ I_3 & 0_3 \end{bmatrix} \begin{bmatrix} F \\ M_P \end{bmatrix}_{\text{ray}} \tag{A.21}$$

Eqs. (A.20) and (A.21) are simply permutations among the screw coordinates. The first three components are changed to the last three components.

The matrix that relates ray and axis-coordinates has some interesting properties[5]

$$\Delta = \begin{bmatrix} 0_3 & I_3 \\ I_3 & 0_3 \end{bmatrix}, \qquad \Delta^T = \Delta, \qquad \Delta^{-1} = \Delta \qquad \det(\Delta) = -1 \tag{A.22}$$

A.3.2 Translation of axes

Another transformation expresses a screw (or screws) in relation to a desired coordinate system. Let firstly this transformation be a simple translation of axes. As discussed in section A.2 this transformation does not change the point independent part of the screw.

In matrix form, the twist transformation in *ray-coordinates* is

$$\begin{bmatrix} \omega \\ v_Q \end{bmatrix}_{\text{ray}} = \begin{bmatrix} I_3 & 0_3 \\ S_{P\to Q} & I_3 \end{bmatrix} \begin{bmatrix} \omega \\ v_P \end{bmatrix}_{\text{ray}} \tag{A.23}$$

[5]Furthermore, Δ gives an indefinite metric for the space of screws. See proof in [Ciblak 1998]. This fact has interesting consequences on the hybrid control problems.

The matrix $S_{P\to Q}$ represents the operator $\overrightarrow{QP}\times$. In other words $S_{P\to Q}x = \overrightarrow{QP}\times x$ for any six dimensional vector x. The matrix $S_{P\to Q}$ is skew-symmetrical.

$$S_{P\to Q} = \begin{bmatrix} 0 & -x_{QP} & y_{QP} \\ x_{QP} & 0 & -z_{QP} \\ -y_{QP} & z_{QP} & 0 \end{bmatrix} \equiv \overrightarrow{QP}\times \tag{A.24}$$

where

$$\overrightarrow{QP} = (x_{QP}, y_{QP}, z_{QP}) = P(x_P, y_P, z_P) - Q(x_Q, y_Q, z_Q)$$

The transpose of $S_{P\to Q}$ is given by

$$S \triangleq S_{Q\to P} = \begin{bmatrix} 0 & -x_{PQ} & y_{PQ} \\ x_{PQ} & 0 & -z_{PQ} \\ -y_{PQ} & z_{PQ} & 0 \end{bmatrix} \simeq \overrightarrow{PQ}\times \tag{A.25}$$

where

$$(x_{PQ}, y_{PQ}, z_{PQ}) = Q(x_Q, y_Q, z_Q) - P(x_P, y_P, z_P)$$

The matrix $\mathbf{T}^{trans}_{Q\to P}$ represents a change of origin of the screw representation without any rotation. In other words $\mathbf{T}^{trans}_{Q\to P}$ corresponds to a *translation* of the coordinate system axes.

$$\mathbf{T}^{trans}_{Q\to P} \triangleq \begin{bmatrix} I_3 & 0_3 \\ S & I_3 \end{bmatrix} \tag{A.26}$$

Naturally

$$\mathbf{T}^{trans}_{P\to Q} = \begin{bmatrix} I_3 & 0_3 \\ S_{P\to Q} & I_3 \end{bmatrix} = \begin{bmatrix} I_3 & 0_3 \\ -S & I_3 \end{bmatrix}$$

as the matrix S in eq. (A.25) is skew-symmetrical.

For wrenches in *ray-coordinates* taken as column vectors similarly

$$\begin{bmatrix} F \\ M_Q \end{bmatrix}_{\text{ray}} = \begin{bmatrix} I_3 & 0_3 \\ S & I_3 \end{bmatrix} \begin{bmatrix} F \\ M_P \end{bmatrix}_{\text{ray}} = \mathbf{T}^{trans}_{P\to Q} \begin{bmatrix} F \\ M_P \end{bmatrix}_{\text{ray}} \tag{A.27}$$

For wrenches in *axis-coordinates taken as column vectors* we have

$$\begin{bmatrix} M_Q \\ F \end{bmatrix}_{\text{axis}} = \begin{bmatrix} I_3 & S \\ 0_3 & I_3 \end{bmatrix} \begin{bmatrix} M_P \\ F \end{bmatrix}_{\text{axis}} = \mathbf{T}^{trans}_{P\to Q} \begin{bmatrix} M_P \\ F \end{bmatrix}_{\text{axis}} \tag{A.28}$$

A.3.3 Rotation of axes

Equations like (A.23) and (A.28) represent the "pointy" variance of the screw representation. This approach when misused can lead to some conceptual mistakes. The whole geometry of the screws is purely *line geometry* and to restrict its aspects to points can be sometimes dangerous as pointed out by Hunt (2000). However the route taken in this section try to explain some frequent doubts for the screw theory newcomer. One of these questions is: why most treatises related to screw theory *define* the Plücker screw coordinates (or the Plücker representation) of a twist with the linear velocity of the point *always* instantaneously coincident with the origin of the coordinate system? When dealing with Statics we cannot make the summation of forces and couple relatively to any point in the space? So why this kind of restriction?

The screw can be represented in relation to *any* point in the space. A triad of coordinate axes can be attached to this point. These coordinate axes normally are self intersecting lines mutually orthogonal in the space. The point of intersection of the three axes is set as the origin of the coordinate system. However this point (the origin of the coordinate system) *need not* to be the point where the summation of forces and couples take place. Naturally a simple translation of

axes (as discussed in section A.3.2) is enough to conciliate both points: the origin of the coordinate system and the point where the screw is refereed to.

In section A.3.2 the aspects of the translation of axes in the screw representation was discussed. Here the effects of a pure rotation of axes are considered. The next step is the conjugation of both changes of coordinates: first a change of the origin (pure translation) and then a (pure) rotation of the frame.

Adding an extra rotation of the coordinate system via a rotation matrix $R_{3\times3}$ expresses each one of the dual vectors in a different coordinate system. Matrix $R_{i\to j}$ is orthogonal and gives the orientation of a free vector specified in frame i, whose origin is Q, in a second frame j, whose origin is P. The coordinate system can be conveniently designated by a single number. This number has a *quid pro quo* relationship with the label that the rigid body receives during the robot analysis. For instance it can be 0 for the ground frame up to n for the end-effector in the Denavit-Hartenberg convention for an n degrees of freedom manipulator. The effect of $R_{i\to j}$ on the screw coordinates can be expressed by the block diagonal matrix

$$\mathbf{T}_{i\to j}^{rot} = \begin{bmatrix} R_{i\to j} & 0_3 \\ 0_3 & R_{i\to j} \end{bmatrix} \tag{A.29}$$

The (complete) screw coordinate transformation matrix ($\mathbf{T}_{i\to j}^{ray}$) is thus expressed in ray-coordinates

$$\mathbf{T}_{i\to j}^{ray} = \begin{bmatrix} R_{i\to j} & 0_3 \\ 0_3 & R_{i\to j} \end{bmatrix} \begin{bmatrix} I_3 & 0_3 \\ S & I_3 \end{bmatrix} = \begin{bmatrix} R_{i\to j} & 0_3 \\ R_{i\to j}S & R_{i\to j} \end{bmatrix} \tag{A.30}$$

or in axis-coordinates

$$\mathbf{T}_{i\to j}^{axis} = \begin{bmatrix} R_{i\to j} & 0_3 \\ 0_3 & R_{i\to j} \end{bmatrix} \begin{bmatrix} I_3 & S \\ 0_3 & I_3 \end{bmatrix} = \begin{bmatrix} R_{i\to j} & R_{i\to j}S \\ 0_3 & R_{i\to j} \end{bmatrix} \tag{A.31}$$

The inverse of matrix $\mathbf{T}_{i\to j}^{ray}$ can be easily derived as

$$\mathbf{T}_{i\to j}^{ray\ -1} = \begin{bmatrix} {R_{i\to j}}^T & S^T {R_{i\to j}}^T \\ 0_3 & {R_{i\to j}}^T \end{bmatrix} = \begin{bmatrix} R_{j\to i} & -SR_{j\to i} \\ 0_3 & R_{j\to i} \end{bmatrix} \tag{A.32}$$

Eq. (A.31) consider wrenches *as column vectors* in axis-coordinates. Wrenches can be expressed as *row vectors* in axis coordinates to take advantage of linear algebra operations. So

$$\hat{\mathbf{T}}_{i\to j}^{axis} = \mathbf{T}_{i\to j}^{axis^T} = \begin{bmatrix} R_{i\to j} & R_{i\to j}S \\ 0_3 & R_{i\to j} \end{bmatrix}^T = \begin{bmatrix} R_{i\to j}^T & S^T R_{i\to j}^T \\ 0_3 & R_{i\to j}^T \end{bmatrix} = \begin{bmatrix} R_{i\to j}^T & -SR_{i\to j}^T \\ 0_3 & R_{i\to j}^T \end{bmatrix} \tag{A.33}$$

APPENDIX B – Chasles' and Poinsot's lines

We intend to demonstrate the existence of Chasles' and Poinsot's lines by a vectorial route.. We need first to demonstrate an important lemma, Lemma 10. The proof follows closely [Ciblak 1998].

The extension of Lemma 10 to Chasles' or Poinsot's Theorems is consequence of the fact that physical quantities in Kinematics or Statics can be represented by screws.

Firstly, a preliminary historical note: neither Chasles nor Poinsot used screw theory to prove their theorems. In reality all screw theory was based on the existence of these theorems as Ball (1900) points out in the introduction of his treatise. Nevertheless the proofs given here, in reverse chronological order, make explicit some screw properties and are interesting by themselves.

B.1 Screw independence on representation

Lemma 10 *The pitch and screw axis of a screw are independent of the point of representation.*

Proof:

Let P and Q be two arbitrary points at which the screws (represented, for instance, in ray order) are given by $[\vec{a}^T \vec{b}_P^T]^T$ and $[\vec{a}^T \vec{b}_Q^T]^T$. This proof will be in

two parts. First, it will be shown that the pitch is invariant and then that the axis is also invariant.

B.1.1 Pitch invariance

From the transformation equation (3.5)

$$\vec{b}_Q = \vec{b}_P + \overrightarrow{QP} \times \vec{a} \tag{B.1}$$

Then by eq. (6.3)

$$h = \frac{\vec{a}^T \vec{b}_Q}{\vec{a}^T \vec{a}} = \frac{\vec{a}^T \vec{b}_P}{\vec{a}^T \vec{a}} + \frac{\vec{a}^T [\overrightarrow{QP} \times \vec{a}]}{\vec{a}^T \vec{a}} = \frac{\vec{a}^T \vec{b}_P}{\vec{a}^T \vec{a}} \tag{B.2}$$

This confirms that the pitch remains the same after changing the point of representation. Using $u = a$ and $v = b_P$ eq. (B.2) is the same as eq. (1.2).

B.1.2 Axis invariance

The screw axis, viewed from the coordinate system at P, is a parameterised *line*, locus of points pointed by a vector r_P. Using eq. (1.3) one possible parameterisation is $r_P(\alpha_P) = \alpha_P \vec{a} + \frac{\vec{a} \times \vec{b}_P}{\vec{a}^T \vec{a}}$. The same line, viewed from the coordinate system at Q, can be parameterised by $r_Q(\alpha_Q) = \alpha_Q \vec{a} + \frac{\vec{a} \times \vec{b}_Q}{\vec{a}^T \vec{a}}$

For the completion of the proof it is only necessary to prove that any point parameterised at P can be parameterised at Q. In other words for any point on the screw axis there are α_P and α_Q such that $r_P(\alpha_P)$ and $r_Q(\alpha_Q)$ locate the same point.

A geometrical argument is to verify whether the vector triangle $r_Q = \overrightarrow{QP} + r_P$ is satisfied for every point on the screw axis or not. If this is true so both

parameterisation lead to the same line (axis). Expanding $\overrightarrow{QP}$

$$\overrightarrow{QP} = r_Q - r_P = (\alpha_Q - \alpha_P)\vec{a} + \frac{\vec{a} \times (\vec{b}_Q - \vec{b}_P)}{\vec{a}^T\vec{a}} \tag{B.3}$$

Using eq. (B.1) into (B.3)

$$\overrightarrow{PQ} = (\alpha_Q - \alpha_P)\vec{a} + \frac{\vec{a} \times (\overrightarrow{PQ} \times \vec{a})}{\vec{a}^T\vec{a}} \tag{B.4}$$

Eq. (B.4) is just a decomposition of vector $\overrightarrow{PQ}$ into two components: one parallel to and other perpendicular to $\vec{a}$. Since this is always possible (and unique), given any α_P it is possible to find an α_Q such that eq. (B.4) is satisfied uniquely. Therefore both representations, at P and at Q, result in the same screw axis. **QED**

B.2 A slightly different approach

Lemma 10 and the argument developed in section B.1 are enough to prove Chasles' or Poinsot's theorems. Another approach is to use directly the screw transformation matrices instead. They show the screw coordinates dependence of the point of representation without considering any rotation of axis.

Both theorems guarantee the existence of a line where the resultant couple and force (in statics case) or the angular and linear velocities of the points (in kinematics case) will be parallel.

Another question is about the existence or not of a vector $r = \overrightarrow{PQ}$ such that the screw transformation becomes

$$\begin{bmatrix} \vec{a} \\ h\vec{a} \end{bmatrix}_{\text{ray}} = \begin{bmatrix} I_3 & 0_3 \\ S & I_3 \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{b}_P \end{bmatrix}_{\text{ray}} \tag{B.5}$$

Here $S \equiv r\times$ as in eq. (4.4).

Eq. (B.5) reduces to a single vector equation

$$h\vec{a} = \vec{b}_P + r \times a \tag{B.6}$$

or

$$\vec{b}_P = h\vec{a} - r \times a = h\vec{a} - r_\perp \times a \tag{B.7}$$

Observe that

$$r \times a = r_\perp \times a$$

where $r_\perp$ is the vector of minimum length defined in Lemma 10, eq. (1.3).

Equation (B.7) derives from the decomposition of r into components parallel to $\vec{a}$ ($h\vec{a}$) and components perpendicular to $\vec{a}$ ($r_\perp \times a$)

$$r = \overrightarrow{PQ} = \alpha\vec{a} + r_\perp \times \vec{a} \tag{B.8}$$

Decompositions like eq. (B.8) are always possible to be done. Vectors r_{perp} and a are mutually perpendicular. The curve generated by the tip of vector r represented by eq. (B.8) is a straight line. This line has α as the free parameter and $\vec{a}/\|\vec{a}\|$ as it versor. This line is the locus that satisfies the related theorem (Chasles' or Poinsot's). **QED**

APPENDIX C – Matrix powers

Let $\bar{F}_{m\times n} = \bar{D}_{m\times r}\bar{E}_{r\times n}$ be a Boolean matrix $\bar{F}$ generated by the product of two generic Boolean matrices $\bar{D}$ and $\bar{E}$. Each element of $\bar{F}$ is a number 0 or 1 and is generated by the summation

$$\bar{F}_{i,j} = \sum_{k=1}^{r} \bar{D}_{i,k} \cdot \bar{E}_{k,j} \tag{C.1}$$

A element $\bar{F}_{i,j}$ is null only if *all* the terms in the summation are zero *i.e.*

$$\bar{D}_{i,k} \cdot \bar{E}_{k,j} = 0 \qquad\qquad \forall k = 1, \ldots, r \tag{C.2}$$

Equations (C.1) and (C.2) can be rewritten in logical (Boolean) terms as

$$\bar{F}_{i,j} = 0 \Longleftrightarrow \bar{D}_{i,k} = 0 \vee \bar{E}_{k,j} = 0 \qquad\qquad \forall k = 1, \ldots, r \tag{C.3}$$

In other words: $\bar{F}_{i,j} = 1$ unless the internal product of row i of first matrix $\bar{D}$ with column j of second matrix $\bar{E}$ is null. This assertion can be explicitly stated as:

"If there is an arc $i \rightarrow k$ in the digraph $G(D)$ so there is not an arc $k \rightarrow j$ in the digraph $G(E)$ and vice-versa"

A important product of two Boolean matrices is the matrix power *i.e.* when $\bar{D} = \bar{E}$ in eq. (C.1). Here both matrices represent the *same* digraph and then

eq. (C.3) indicates that there is no path of length 2 from vertex j to vertex i in the graph represented by the incidency matrix $\bar{D}$ $(= \bar{E})$. The same assertion above can now be restated:

"If there is an arc $i \rightarrow k$ in the digraph $G(D)$ so there is not an arc $k \rightarrow j$ in the *same* digraph $G(D)$ and vice-versa. So its is impossible to reach vertex j starting from vertex i when $F_{i,j} = 0$"

An incidency matrix $\bar{A}$ is the first power of itself: $\bar{A} = \bar{A}^1$. An "1" value in an element of the matrix $\bar{A}$ imply the existence of a path of length 1. A simple example will illustrate this result.

Let $\bar{A}^2 = \bar{A} \cdot \bar{A}$ using the example of Figs. 2 and 3

$$\bar{A}^2 = \bar{A} \cdot \bar{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{C.4}$$

There is only one null element on $\bar{A}^2$, eq. (C.4), *i.e.* $a_{32}^{(2)} = 0$. The superscript "(2)" in $a_{32}^{(2)}$ indicates the power of matrix $\bar{A} = [a_{ij}]$. Starting from vertex 2 is impossible to reach vertex 3 with a path of length exactly 2. This is the only impossible path $i \rightarrow j$ with length 2 between any arbitrary pair of vertices (i, j) in the graph represented by Fig. 2. However there are paths $2 \rightarrow 3$ with lengths different from 2 as *e.g.* the single arc (path with length 1) $2 \rightarrow 3$.

Two useful lemmas related to matrix powers are [Reinschke 1988] [Jantzen 1996]

Lemma 11 *Let B^k be a $k-$power of the incidency matrix B of a directed graph and $b_{ij}^{(k)}$ an element of B^k then $b_{ij}^{(k)} = 1$ only if there is a path with length k starting from vertex j and ending at vertex i in the graph represented by the incidency matrix B, otherwise $b_{ij}^{(k)} = 0$.*

Sketch of the proof: As $\bar{A}^2 = \bar{A}\bar{A}$ represents paths with length 2 then $\bar{A}^3 = \bar{A}^2\bar{A} = \bar{A}\bar{A}^2$ represents a length 3 path and so on.

Another useful Lemma is

Lemma 12 *Let B^k be a $k-$power of the incidency matrix $B_{n\times n}$ of a directed graph with n vertices. If $k \geq n$ then B^k will be a full Boolean matrix or will repeat cyclically the patterns of the powers $B^1, B^2, \ldots, B^{n-1}$.*

Sketch of the proof: Every path with length greater than or equal to n will necessarily pass twice by at least one vertex of the graph. If there is such a path passing through every vertex of the graph then the reachability matrix $\bar{R}$ is full. Otherwise, these paths will circulate the strong components of the graph.

Formal proofs of Lemmas 11 and 12 are found in [Reinschke 1988]. The conjugation of both lemmas above leads to eq. (2.4) which obtains the reachability matrix from the incidency matrix $\bar{R} = \sum_{i-1}^{n-1} \bar{A}^k$.

APPENDIX D – Numerical computation of the inverse Jacobian matrix structure

The kinematic structure matrix Q_{kin} depends upon the structure of the Jacobian matrix *and* its inverse. The Jacobian matrix can be symbolically derived in a straightforward manner either using Denavit-Hartenberg parameters and their coordinate systems or using screw theory.

However the symbolic computation of the inverse Jacobian matrix is a tougher task [Lucas, Tischler e Samuel 2000]. Some hierarchical kinematic analysis advantages, which depends upon the kinematic structure matrix, would be lost if one must compute Q_{kin} for each robot architecture. Moreover a single robot architecture can have as many Jacobian matrices as coordinate systems and variables (angles or lengths) are chosen to describe it.

The efficiency of the analysis of a specific robot architecture is, in a great extent, a quest for a good coordinate system (possibly the best one) and a corresponding set of descriptor variables that turn the Jacobian as simple as possible. If, for every trial in the robot analysis, the Jacobian matrix has to be symbolically inverted the analysis procedure becomes quite cumbersome.

Procedurally Q_{kin} can be expressed only by its structure matrix $\bar{Q}_{kin}$ *i.e.* there is no need to know *explicitly* the real values of the elements of J and J^{-1}. The

crucial point is the existence (or not) of a null element of the matrices J and J^{-1} at a certain position.

Separation between null and non null elements in the inverse of the Jacobian matrix can be done by purely numerical or graphical methods so the Jacobian matrix need not to be symbolically inverted. One procedure is to create auxiliary graphs for each element of the original matrix, see [Jantzen 1996]. Other procedure is based upon the inverse of randomly generated realizations of the Jacobian matrix and in the observation of the pattern of the elements of the inverse matrix.

The latter was adopted in this work due to the availability of computational resources, particularly the software Octave [Eaton 1996]. The method consists in, given a structure matrix, fill the indeterminate (non-null) values with a randomly generated number and then observe the behaviour of the inverse of this matrix. After some trials the structure of the inverse matrix is obtained *i.e.* the null terms of the inverse are located.

Due to the numerical inaccuracy of this method a chosen threshold $\epsilon > 0$ is used to decide whether the element is properly a zero term contaminated by the propagation of rounding errors and binary representation or it is a non-null function of q. Each element is compared, in module, with this threshold $\|q_{ij}\| < \epsilon$. In this case $q_{ij} \in Q_{kin}$ is an element of the kinematic structure matrix. The choice of ϵ depends upon the robot architecture and the analyst's criteria.

Algorithm 2 describes how to obtain the structure of the inverse Jacobian numerically. Once $\overline{J^{-1}}$ is obtained, matrix $\bar{Q}_{kin}$ can be mounted using algorithm 2:

$$\bar{Q}_{kin} = \begin{pmatrix} 0 & \bar{J} \\ \overline{J^{-1}} & 0 \end{pmatrix} \tag{D.1}$$

Algorithm 2: Structure of the inverse matrix obtained numerically

ALGORITHM: **Input data:** Jacobian matrix J

1. Extract the structure matrix $\langle J \rangle$
2. Compute $\tilde{J} = J(q)$ with arbitrary values for the variables q_i, or
3. If $J = J(q)$ is not available,generate a realization $\tilde{J} \in \langle J \rangle$ with random values
4. Invert $\tilde{J}$ and check the values, in module, with the threshold ϵ.
5. Repeat items 3, 2 and 4 to confirm the structure matrix of J^{-1} ie $\langle J^{-1} \rangle$

APPENDIX E – Kirchhoff-Davies circuit law

E.1 Adaptation of the Kirchhoff laws to Kinematics

As parallel robots are closed kinematic chains we need to know how to solve the kinematic problem. The most general technique to solve the kinematics of closed kinematic chains appears to be the Kirchhoff-Davies laws [Davies 1983] [Davies 1981] [Davies 1995] [Davies 2000] [Campos 2001], which will be described in this section.

The Kirchhoff's circuit law is a fundamental result to solve circuits in electrical engineering. Davies (1981) adapted this law to the kinematics of closed kinematic chains, combining graph theory and screw theory. This adaptation solves the set of instantaneous screws transmitted by independent kinematic pairs to the other pairs of the chain in a given architecture. The method leads to a matricial formulation that, additionally, facilitates computational solutions and detects singularities.

The method is quite general, as it is based on the topology of the kinematic chain. Kirchhoff-Davies circuit law generates the kinematic matrices by graph theoretical concepts. The method will be applied to a planar platform with three degrees of freedom $3RRR$.

Initially, the platform kinematics is explained by constructing the coupling graph of the chain, meanwhile some additional concepts of the theory of graphs are explained. In the sequence, the movement graph of the kinematic chain is generated. Finally, by means of the adaptation of the Kirchhoff's circuit law, the matrix containing the instantaneous screws of all the kinematic pairs of the closed chain is calculated.

An additional special terminology described in [Davies 1995] is used to deal with kinematics chains and graphs.

E.2 Coupling graph of a kinematic chain

The coupling graph G_C of a kinematic chain represents each link of the chain by means of one *vertex* n (identified by a number) and each kinematic pair of the chain by means of an *edge* e (identified by a letter). If an orientation is assigned to each edge of the graph, we obtain a digraph of the kinematic chain as discussed in section 2.2.1 and these edges are called arcs. The orientation of the arcs can be arbitrarily chosen, such as from vertices with lower numbers to vertices with higher numbers. The mechanism $3RRR$ in Fig. 37, with $n = 8$ and $e = 9$, has an associated coupling graph shown in Fig. 38.

Some extra concepts of the theory of graphs, not mentioned in Chapter 2, are here defined, or redefined, in conformity with [Davies 1995]:

- Circuit: a synonym of closed path *i.e.* a path in which the initial and final node are the same, see definitions in section 2.2.1 at page 2.2.1.
- Tree: a subgraph which contains all vertices of the graph but no circuits. A graph can have several associated trees.

  Some authors [Harary, Norman e Cartwright 1965] [Tsai 2001] define a tree as *any* subgraph containing no circuits. A tree involving all the vertices of the original graph is, by these authors, called a *spanning tree*.

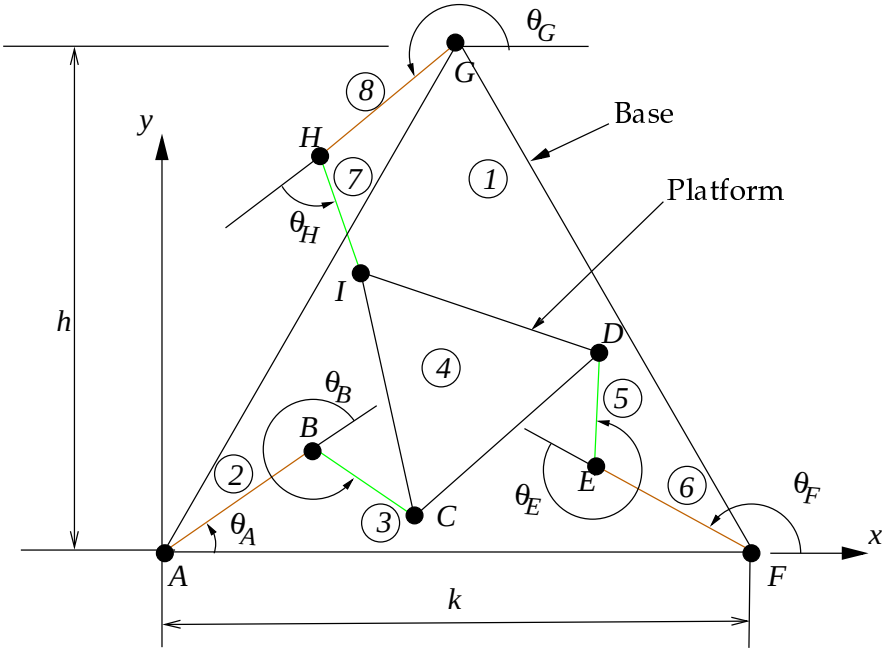

Figure 37: Mechanism $3RRR$ in the xy-plane

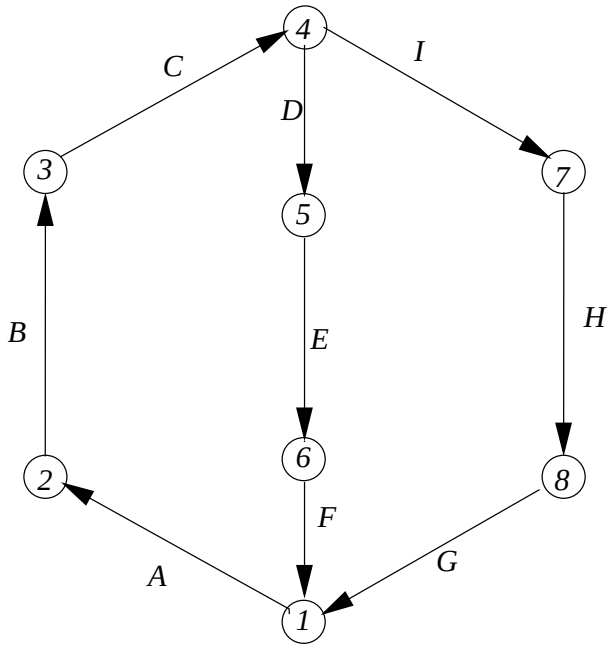

Figure 38: Graph of mechanism $3RRR$.

- Branches: arcs of the graph contained in a given (spanning) tree.
- Chords: the arcs which were omitted of the graph in a preselected (spanning) tree. For each chord of the graph G_C exists a circuit which, in general, takes the name and the orientation (clockwise or counter-clockwise) determined by the corresponding chord. The number of chords equals the number of circuits in a graph.
- Cutset: the set of arcs composed by a unique branch and the minimum number of chords which, when separated from the graph, split the graph into two disjoint subgraphs. There is only one cutset for each branch of a preselected tree. Furthermore, the chords in this cutset are uniquely determined by the arcs lacking from the circuits where the branch pertains to. Notice that the number of cutsets equals the number of the branches in a graph.

The incidency matrix discussed in chapter 2 is a form of matricial representation of digraphs, but it not the only one. Another form, maybe more suitable to represent the circuits inside a digraph like G_C, is the matrix $[B]_{(l\times e)}$ where l indicates the number of circuits (rows of $[B]$) and e the number of arcs (columns of $[B]$). Each element (b_{ij}) of $[B]_{(l\times e)}$ is:

- 0, if the circuit i does not include the arc j,
- $+1$, if the positive orientation of the circuit i is the same orientation of the arc j included in this circuit, and
- -1, if the positive orientation of the circuit i is the opposite of the orientation of arc j included in this circuit.

For the mechanism $3RRR$, the chords were chosen as the arcs A and G, which yields $l = 2$ circuits. Figure 39 shows chords A and G (with thick lines) and circuits M_A and M_G.

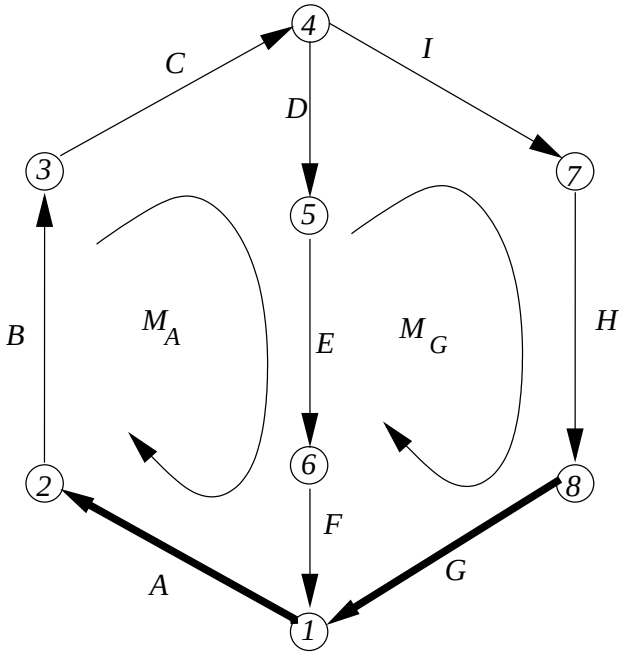

Figure 39: Chords and circuits of the graph G_C for the mechanism $3RRR$

Thus, the matricial representation of the two circuits M_A and M_G of G_C is:

$$B = \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \left[\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 \end{array}\right] \end{array} \begin{array}{c} M_A \\ M_G \end{array} \tag{E.1}$$

E.3 Movement graph of a kinematic chain

The movement[1] graph G_M is a coupling graph where the arcs represent kinematic pairs allowing only $f = 1$ degree of freedom. Thus, each one of the arcs of G_C, representing direct kinematic pairs with $f > 1$ degrees of freedom, are substituted by f arcs representing direct kinematic pairs with $f = 1$ degree of

[1]While Davies (2000) uses the term *motion graph*, we adopt here the nomenclature suggested by Hunt (1978) and corroborated by the Webster's dictionary which asserts that: *Motion expresses a general idea of not being at rest; movement is oftener used to express a definite, regulated motion, especially a progress.*. In this sense, motion could involve *e.g.* accelerations, inertia factors, etc ... which were clearly outside the scope of this work.

freedom. These substituted arcs are placed in series and, by convenience, with the same orientation of the original arc.

Between the substitute arcs appear $f - 1$ fictitious vertices (links) with the only objective to mediate these two bodies, which were directly connected by the original kinematic pair. Each set of f arcs of G_M, which represents an arc of G_C, defines a single screw. Such screws, as a whole, determine the f-system of screws [Hunt 1978] allowed by the direct kinematic pair represented by an arc of G_C. The f-system of the direct kinematic pair between two links is, in general, the screw system of twists that one link has relative to the other if they were joined directly, as *e.g.* in the case of the serial manipulator.

Figure 40 shows the movement graph G_M for the mechanism $3RRR$. Notice that the substitution of arcs is not observed, as each one of the e arcs has $f = 1$ degree of freedom.

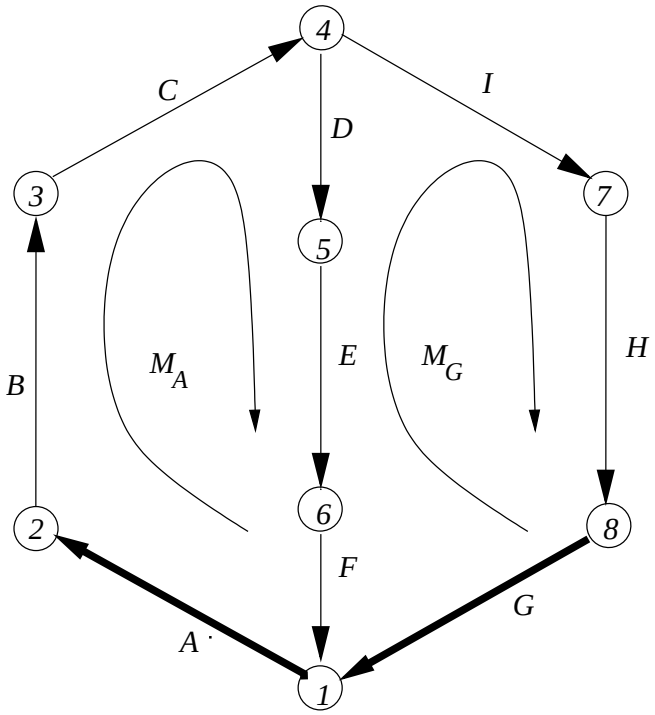

Figure 40: Chords and circuits of the graph G_M for the mechanism $3RRR$

The indirect unions on a closed kinematic chain impose more restrictions

on the movement than the direct kinematic pairs, thus the degree of freedom f of the directly connected links i and j can be reduced to f_{ij}, $f_{ij} \leq f_i$ and $f_{ij} \leq f_j$, due to the presence of the other kinematic pairs. We can define the gross degree of freedom F of a kinematic chain as the summation $F = \sum_1^e f_i$ of all arcs of G_M where each arc represents one degree of freedom. For the mechanism $3RRR$ the gross degree of freedom is calculated from the kinematic pairs A, B, C, D, E, F, G, H and I:

$$F = \sum_1^9 f_i = \sum_1^9 1 = 9 \tag{E.2}$$

Similarly to the graph G_C, the graph G_M also possesses a matrix which represents the circuits of the graph: $[B_M]_{(l \times F)}$, where $F \leq e$, *i.e.* the number of arcs in G_M cannot be lower than the number of arcs in G_C.

Rewriting the matrix $[B]$, but repeating f_i times the columns of $[B]$ corresponding to the substituted arcs, leads to a matricial representation $[B_M]$ of the circuits of G_M. For convenience, again, these substituted arcs will keep the same original of the original arc.

As discussed above, the example discussed here has no kinematic pairs with $f_i > 1$, and then the matricial representation $[B_M]$ of the circuits of G_M is the same as that of $[B]$:

$$B_M = B = \begin{array}{c} \begin{array}{ccccccccc} A & B & C & D & E & F & G & H & I \end{array} \\ \left[\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 1 & 1 & 1 \end{array}\right] \end{array} \begin{array}{c} \\ \begin{array}{c} M_A \\ M_G \end{array} \end{array} \tag{E.3}$$

E.4 Kirchhoff-Davies circuit law

The Kirchhoff's circuit law adapted to the kinematic case requires the relationship of the screws of all the kinematic pairs pertaining to the same circuit. The basic principle underlining this adaptation is that the movement of a link relatively to itself is always null. Thus, the matrix of the normalised screws (in open

chain) of the kinematic pairs $\hat{M}_D$ is constrained by the the matrix that relates the role each link has in each circuit B_M. The matrix of normalised screws $\hat{M}_N$ combines these two aspects of the closed kinematic chains, namely: independent screws associated with each joint and the closure of the kinematic chain.

Let d ($1 \leq d \leq 6$) be the minimum order of the screw-system (d-system) to which belong all the F screws of the kinematic chain. So, we generate the matrix $[\hat{M}_D]_{(d\times F)}$ which represents all (normalised) screws of the direct kinematic pairs, *i.e.* the screws allowed for each kinematic pair without the restrictions of the remaining kinematic pairs of the kinematic chain. Each column of $[\hat{M}_D]_{(d\times F)}$ is the normalised screw of an arc of G_M (one degree-of-freedom screw).

As the mechanism $3RRR$ is planar, $d = 3$, we consider only three Plücker's coordinates $\mathcal{N}, \mathcal{P}$ and $\mathcal{Q}$ [Hunt 1978]. Therefore, the matrix of normalised screws, where the d rows represent the Plücker's coordinates and the F columns represent the instantaneous screw at each kinematic pair of $f = 1$ degree of freedom, is given by

$$\hat{M}_D^T = \begin{array}{c} \begin{matrix} \mathcal{P} & \mathcal{Q} & \mathcal{N} \end{matrix} \\ \left[\begin{array}{ccc} 0 & 0 & 1 \\ ps_a & -pc_a & 1 \\ qs_{ba} + ps_a & -qc_{ba} - pc_a & 1 \\ qs_{fe} + ps_f & -qc_{fe} - pc_f - k & 1 \\ ps_f & -pc_f - k & 1 \\ 0 & -k & 1 \\ hk & -\frac{k}{2} & 1 \\ ps_g + hk & -pc_g - \frac{k}{2} & 1 \\ qs_{hg} + ps_g + hk & -qc_{hg} - pc_g - \frac{k}{2} & 1 \end{array}\right] \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{matrix} \end{array} \tag{E.4}$$

where the matrix $\hat{M}_D$ is shown transposed in eq. (E.4) due to space restrictions. In eq. (E.4) $s_a = \sin\theta_A$, $s_b = \sin\theta_B$, $c_a = \cos\theta_A$, $s_{ba} = \sin(\theta_B+\theta_A)$, $c_{ba} = \cos(\theta_B+\theta_A)$, etc ... The dimensions $p = \overline{AB} = \overline{EF} = \overline{GH}$, and $q = \overline{BC} = \overline{FG} = \overline{HI}$ complete the dimensions shown in Fig. 37.

Matrix $\hat{M}_D$ does not represent the true plot of the movements of the kinematic

chain and cannot be used directly to obtain the kinematic equations of the chain. The information about the closure of the chain is given by matrix B_M in eq. (E.3).

Due to the restrictive connection among the kinematic pairs, it is necessary to find a matrix of normalised screws for the kinematic chain as whole. This matrix must implicitly contain all information about the twists on the kinematic pairs including the constraint imposed by the remaining kinematic pairs.

From the graph G_M of the mechanism $3RRR$, Fig. 40, we observe that exists $l = 2$ circuits: M_A and M_G. On the other hand, the Kirchhoff-Davies circuit law requires, for each circuit, that the summation of the $d = 3$ components be zero. This indicates that the movement of a link in relation to itself is null even when this movement is expressed by the intermediate movements of all the kinematic pairs of the circuit. So the Kirchhoff-Davies circuit law gives dl equations.

To obtain the matricial form of the Kirchhoff-Davies circuit law, it is necessary to extract l diagonal matrices of $[B_M]_{(l\times F)}$ of the form $[B_i]_{(F\times F)}$, $i = 1, 2, \ldots, l$, where the diagonal elements of $[B_i]_{(F\times F)}$ are the elements of the row i of $[B_M]_{(l\times F)}$. Therefore, the normalised screw matrix of the manipulator $[\hat{M}_N]_{(dl\times F)}$ is [Davies 2000]:

$$[\hat{M}_N]_{(dl\times F)} = \begin{bmatrix} \left[\hat{M}_D\right]_{(d\times F)} [B_1]_{(F\times F)} \\ \left[\hat{M}_D\right]_{(d\times F)} [B_2]_{(F\times F)} \\ \left[\hat{M}_D\right]_{(d\times F)} [B_3]_{(F\times F)} \\ \cdots \\ \left[\hat{M}_D\right]_{(d\times F)} [B_l]_{(F\times F)} \end{bmatrix}_{(dl\times F)} \tag{E.5}$$

and the Kirchhoff-Davies circuit law becomes

$$[\hat{M}_N]_{(dl\times F)}[\Psi]_{(F\times 1)} = [0]_{(dl\times 1)} \tag{E.6}$$

where $[\Psi]_{(F\times 1)}$ is the vector of magnitudes of the twists *i.e.* angular and linear velocities.

Matrix $\hat{M}_N$ for the mechanism $3RRR$ is:

$$\hat{M}_N = \begin{bmatrix} 0 & p\,s_a & q\,s_{ba} + p\,s_a & q\,s_{fe} + p\,s_f \\ 0 & -p\,c_a & -q\,c_{ba} - p\,c_a & -q\,c_{fe} - p\,c_f - k \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -q\,s_{fe} - p\,s_f \\ 0 & 0 & 0 & q\,c_{fe} + p\,c_f + k \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdots \begin{bmatrix} p\,s_f & 0 & 0 & 0 & 0 \\ -p\,c_f - k & -k & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ -p\,s_f & 0 & H\,k & p\,s_g + H\,k & q\,s_{hg} + p\,s_g + H\,k \\ p\,c_f + k & k & -\frac{k}{2} & -p\,c_g - \frac{k}{2} & -q\,c_{hg} - p\,c_g - \frac{k}{2} \\ -1 & -1 & 1 & 1 & 1 \end{bmatrix} \tag{E.7}$$

and the vector of magnitudes is represented by the angular velocities at the kinematic pairs:

$$\Psi = \begin{bmatrix} \Psi_A & \Psi_B & \Psi_C & \Psi_D & \Psi_E & \Psi_F & \Psi_G & \Psi_H & \Psi_I \end{bmatrix}^T \tag{E.8}$$

where Ψ_i corresponds to the magnitude of the velocity at the kinematic pair i.

The system $[\hat{M}_N][\Psi] = [0]$ defines completely the kinematic problem. To solve it, one must select the joints which will "carry on" the primary variables and those joints which will "carry on" the secondary variables. The primary variables are the joint velocities imposed by the actuators while the secondary variables are the velocities at the so-called passive joints.

References

ANGELES, J. *Fundamentals of Robotic Mechanical Systems*. New York: Springer, 1997.

ARTOBOLEVSKI, I. I. Теория механизмов и машин. fourth. Moscow: Nauka, 1988. 640 p. In English *Mechanism and Machine Theory*. ISBN 5-02-013810-X.

ASADA, H.; SLOTINE, J. J. E. *Robot Analysis and Control*. [S.l.]: John Wiley and Sons, 1986.

ASSUR, L. V. Исследование плоских стерневых механизмов с низими парами с токи зрения их структуры и классификации. [S.l.]: Izdat. Akad. Nauk SSSR, 1952. 592 pp. (4 folded insets, 1 plate) p. Edited by I. I. Artobolevskiĭ. In English *Research about the structure and classification of plane mechanisms*.

BAILLIEUL, J. Kinematic programming alternatives for redundant manipulators. In: *Proc. 1985 IEEE Int. Conf. Robotics & Automation*. [S.l.: s.n.], 1985. p. 722–728.

BAKER, D.; WAMPLER, C. The inverse kinematics of redundant manipulators. *The International Journal of Robotics Research*, v. 7, n. 2, p. 3–21, March/April 1988.

BALL, R. S. *A Treatise on the Theory of Screws*. Cambridge: Cambridge University Press, 1900. ISBN 0521636507 -reedicao 1998.

BARANOV, G. G. *Curso de la Teoría de Mecanismos y Máquinas*. second. moscow: mir, 1985. Spanish translation. Russian original.

BOTTEMA, O.; ROTH, B. *Theoretical Kinematics*. New York: Dover, 1979.

BRUALDI, R. A.; RYSER, H. J. *Combinatorial matrix theory*. Cambridge [England] ; New York: Cambridge University Press, 1991. ISBN 0521322650.

CAMPOS, A. *Cálculo da Matriz Cinemática de Rede do Manipulador Paralelo 3RRR baseado na Lei de Kirchhoff e na Teoria de Helicóides*. Florianópolis, SC, 2001.

CHANG, P. H. *A Closed Form Solution for Inverse Kinematics of Robot Manipulator with Redundancy*. [S.l.], 1986.

CHIAVERINI, S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, IEEE, p. 399–410, July 1997.

CIBLAK, N. *Analysis of Cartesian Stiffness and Compliance with Applications*. Tese (Doutorado) — Georgia Institute of Technology, maio 1998.

COE, C. J. *Theoretical mechanics : a vectorial treatment*. New York: The Macmillan, 1938.

CORKE, P. I. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, v. 3, n. 1, p. 24–32, Mar 1996.

DAVIES, T. The 1887 committee meets again. subject: freedom and constraint. In: HUNT, H. (Ed.). *Ball 2000 Conference*. Trinity College: Cambridge University Press, 2000. p. 1–56.

DAVIES, T. H. Kirchhoff's circulation law applied to multi-loop kinematic chains. *Mechanism and Machine Theory*, v. 16, p. 171–183, 1981.

DAVIES, T. H. Mechanical networks—I: Passivity and redundancy. *Mechanism and Machine Theory*, v. 18, n. 2, p. 95–101, 1983.

DAVIES, T. H. Mechanical networks—III: Wrenches on circuit screws. *Mechanism and Machine Theory*, v. 18, n. 2, p. 107–112, 1983.

DAVIES, T. H. Circuit actions attributable to active couplings. *Mechanism and Machine Theory*, v. 30, n. 7, p. 1001–1012, 1995.

DAVIES, T. H. Couplings,coupling network and their graphs. *Mechanism and Machine Theory*, v. 30, n. 7, p. 991–1000, 1995.

DENAVIT, J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. ASME E, Journal of Applied Mechanics*, v. 22, p. 215–221, June 1955.

DIMENTBERG, F. M. A general method for the investigation of finite displacements of spatial mechanisms and certain cases of passive joints. *Akad. Nauk, SSSR Trudi Sem. Teorii Mash. Mekh.*, v. 5, n. 17, p. 5–39, Purdue Translation No. 436 1948.

DOTY, K. L.; MELCHIORRI, C.; BONIVENTO, C. A theory of generalized inverse applied to robotics. *Int. J. Robotics Research*, v. 12, n. 1, p. 1–19, 1993.

DUFFY, J. The fallacy of modern hybrid control theory that is based on 'orthogonal complements' of twist and wrench space. *Journal of Robotic Systems*, v. 7, n. 2, p. 139–144, 1990.

EATON, J. W. *Octave: A High Level Interactive Language for Numerical Computations*. Second. [S.l.], October 1996. For Octave Version 1.94.

FANGHELLA, P.; GALLETTI, C. Kinematics of robot mechanisms with closed actuating loops. *International Journal of Robotics Research*, v. 9, n. 6, p. 19–24, Dec 1990.

FANGUELLA, P. Kinematics of single-loop mechanisms and serial robot arms: A systematic approach. *Meccanica*, v. 30, p. 685–705, 1995.

FANGUELLA, P.; GALLETTI, C. Particular or general methods in robot kinematics. *Mechanism and Machine Theory*, v. 24, n. 5, p. 383–394, 1989.

GIBSON, C. G.; HUNT, K. H. Geometry of screws systems parts 1 and 2. *Mechanism and Machine Theory*, v. 25, n. 1, p. 1–27, 1990.

GOLUB, G. H.; LOAN, C. F. van. *Matrix Computations*. [S.l.]: North Oxford Academic Publishing, 1983.

GOSSELIN, C. M. Kinematische und statische analyse eines ebenen parallelen manipulators mit dem freihaitsgrad zwei. *Mechanism and Machine Theory*, v. 31, n. 2, p. 149–160, 1996.

GOSSELIN, C. M.; MERLET, J.-P. The direct kinematics of planar parallel manipulators: Special architectures and number of solutions. *Mechanism and Machine Theory*, v. 29, n. 8, p. 1083–1097, 1994.

GOUGH, V. E.; WHITEHALL, S. G. Universal tyre testing machine. In: F.I.S.I.T.A, INSTITUTION OF MECHANICAL ENGINEERS. *Proceedings of the 9th International Technical Congress*. London, 1962.

GUENTHER, R.; SIMAS, H.; PIERI, E. R. de. Concepção cinemática de um manipulador para volumes de trabalho restritos. In: *Proceedings of the Congresso Nacional de Engenharia Mecânica*. [S.l.: s.n.], 2000. in CDROM. In English : *Kinematic conception of a manipulator for constrained workspaces*.

HARARY, F. (Ed.). *Graph Theory and Theoretical Physics*. London: Academic Press, 1967.

HARARY, F.; NORMAN, R. Z.; CARTWRIGHT, D. *Structural Models: An Introduction to the Theory of Directed Graphs*. 4. ed. New York: John Wiley & Sons, 1965.

HARTENBERG, R. S.; DENAVIT, J. *Kinematic Synthesis of Linkages*. New York: McGraw-Hill, 1964.

HOLLERBACH, J. M. Optimum kinematic design for a seven degree of freedom manipulator. In: HANAFUSA, H.; INOUE, H. (Ed.). *Robotics Research 2nd International Symposium*. Kyoto, Japan: MIT Press, 1985. p. 215–222.

HU, Y.-R.; GOLDENBERG, A. A. An adaptive approach to motion and force control of multiple coordinated manipulators. *Journal of Dynamic Systems, Measurement, and Control*, v. 115, p. 60–69, 1993.

HU, Y.-R.; VUKOVICH, G. Dynamic control of free-floating coordinated space robots. *Journal of Robotic Systems*, v. 15, n. 4, p. 217–230, 1998.

HUNT, K. H. *Kinematic Geometry of Mechanisms*. Oxford: Clarendon Press, 1978.

HUNT, K. H. Structural kinematics of in-parallel-actuated robot-arms. *Trans. ASME, Journal of Mechanisms, Transmissions and Automation in Design*, v. 105, p. 705–712, Dec 1983.

HUNT, K. H. Robot kinematics—a compact analytic inverse solution for velocities. *Trans. ASME, Journal of Mechanisms, Transmissions and Automation in Design*, v. 109, p. 42–49, March 1987.

HUNT, K. H. Dont't cross-thread the screw. In: HUNT, H. (Ed.). *Ball 2000 Conference*. Trinity College: Cambridge University Press, 2000. p. 1–37.

HUNT, K. H.; MCAREE, P. R. The octahedral manipulator: Geometry and mobility. *International Journal of Robotics Research*, v. 17, n. 8, p. 868–885, 1998.

INNOCENTI, C. Polynominal solution to the position analysis of the 7-link assur kinematic chain with one quaternary link. *Mechanism and Machine Theory*, v. 30, n. 8, p. 1295–1303, 1995.

JANTZEN, J. *Digraph Analyses of Linear Control Systems*. Dept of Automation: Technical University of Denmark, 1996. Publ. 96-H-838 (lecture notes).

JR, M. B. L.; NUGENT, L. M.; SARIDIS, G. N. Efficient PUMA manipulator Jacobian calculation and inversion. *Journal of Robotic Systems*, v. 4, n. 2, p. 185–197, 1987.

KATZ, A. *Computational Rigid Vehicle Dynamics*. first. Malabar, Florida, USA: Krieger Publisher Company, 1997.

KLEIN, C.; CHU-JENQ, C.; AHMED, S. Use of an extended jacobian method to map algorithmic singularities. In: Werner, Robert; O'Conner, L. (Ed.). *Proceedings of the 1993 IEEE International Conference on Robotics and Automation: Volume 3*. Atlanta, GE: IEEE Computer Society Press, 1993. p. 632–637. ISBN 0-8186-3450-2.

KLEIN, C.; CHU-JENQ, C.; AHMED, S. *A new formulation of the extended Jacobian method and its use in mapping algorithmic singularities for kinematically redundant manipulators*. 1995.

KLEIN, C. A.; HUANG, C. H. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems Man and Cybernetics*, v. 13, n. 3, p. 245–250, 1983.

KU, D.-M. Direct displacement analysis of a Stewart platform mechanism. *Mech. Mach. Theory*, v. 34, n. 3, p. 453–465, 1999.

LUCAS, S. R.; TISCHLER, C. R.; SAMUEL, A. E. Real-time solution of the inverse kinematic-rate problem. *International Journal of Robotics Research*, v. 19, n. 12, p. 1236–1244, December 2000.

MANOLESCU, N. I. For a united point of view in the study of the structural analysis of kinematic chains and mechanisms. *Journal of Mechanisms*, v. 3, p. 149–169, 1968.

MANOLESCU, N. I.; MANAFU, V. Sur la determination du degré de mobilité des mécanismes. *Bull Polytech. Inst. Bucharest*, v. 25, p. 45–66, 1963.

MARTINS, D. *Geometrical and Combinatorial Approaches to Robot Manipulators Kinematics*. Florianópolis, SC, 2000.

MARTINS, D.; GUENTHER, R. Hierarchical singularity analysis of an articulated robot. In: *Proceedings of the Brazilian Congress of Mechanical Engineering -IX COBEM*. Uberlandia: Brazilian Society of Mechanical Sciences, 2001. p. ??–??

MARTINS, D.; GUENTHER, R. Hierarchical singularity analysis of robots. In: ESPINDOLA, J. J.; LOPES, E. M. O.; BAZAN, F. S. V. (Ed.). *Proceedings of the Nineth International Symposium on Dinamic Problems in Mechanics -IX DINAME*. Florianopolis: Brazilian Society of Mechanical Sciences, 2001. p. 219–224. ISBN 85-85769-05-X.

MARTINS, D.; MACIE, C. A. Sistemática computacional para a resolução de grupos de Assur de segunda ordem. In: MACHADO, R. D. (Ed.). *Anais do XIV CILAMCE -Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia*. [S.l.], 1995. v. 2. In Portuguese.

MARTINS, D.; MACIE, C. A. Determinação de reações em grupos de Assur de primeira ordem. In: UNIVERSIDAD DE VALDIVIA. *Congreso Nacional Chileno de Ingeniería*. Valdivia, Chile, 1996.

MARTINS, D.; MACIE, C. A. Determinação de reações em grupos de Assur de primeira ordem. In: UNIVERSIDAD DE VALDIVIA. *Congreso Nacional Chileno de Ingeniería*. Valdivia, Chile, 1996.

MARTINS, D.; SIMAS, H.; GUENTHER, R. *Singularity avoidance based on the globally constrained Jacobian method*. Florianópolis, SC, 2001. In Portuguese.

MOZZI, G. *Discorso matematico sopra il rotamiento momentaneo dei corpi*. naples: Stamperia di Donato Campo, 1763.

MUROTA, K. *Systems analysis by graphs and matroids: Structural solvability and controllability*. Berlin: Springer-Verlag, 1987. ISBN 3-540-17659-4.

MUROTA, K. *Matrices and Matroids for Systems Analysis*. Berlin: Springer-Verlag, 2000. ISBN 3540660240.

MUROTA, K.; SCHARBRODT, M. Computing the combinatorial canonical form of a layered mixed matrix. *Optimization Methods and Software*, v. 10, n. 2, p. 373–391, 1998.

MURRAY, R. M.; LI, Z.; SASTRY, S. S. *A Mathematical Introduction to Robotic Manipulation*. Ann Arbor: CRC Press, 1994.

NAKAMURA, Y.; HANAFUSA, H. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement and Control*, v. 108, p. 163–171, September 1986.

NENCHEV, D. N. Redundancy resolution through local optimization: A review. *J. Robot. Syst.*, v. 6, n. 6, p. 769–798, December 1989.

ORWANT, J.; HIETANIEMI, J.; MACDONALD, J. *Mastering Algorithms with Perl*. [S.l.]: O'Reilly & Associates, 1999. ISBN 1-56592-398-7.

PARENTI-CASTELLI, V.; GREGORIO, R. D. Closed-form solution of the direct kinematics of the 6-3 type stewart platform using one extra sensor. *Meccanica*, v. 31, n. 6, p. 705–714, 1996.

PARK, J.; CHUNG, W. K.; YOUM, Y. Behaviors of extended jacobian method for kinematic resolutions of redundancy. In: STRAUB, E.; SIPPLE, R. S. (Ed.). *Proceedings of the International Conference on Robotics and Automation. Volume 1*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994. p. 89–95. ISBN 0-8186-6061-9.

PHILLIPS, J. *Freedom in Machinery, Vol 1*. Cambridgeshire: Cambridge University Press, 1984.

PHILLIPS, J. *Freedom in Machinery, Vol 2*. Cambridgeshire: Cambridge University Press, 1990.

PISSANETSKY, S. *Sparse Matrix Technology*. New York, NY, USA: Academic Press, 1984. xiii + 321 p. ISBN 0-12-557580-7.

PLüCKER, J. On a new geometry of space. *Phil. Trans.*, v. 155, p. 725–791, 1865.

POINSOT, L. Sur la composition des moments et la composition des aires. *J. Éc Polyt. Paris*, v. 6, p. 182–205, 1806.

REINSCHKE, K. J. *Multivariable control : a graph-theoretic approach*. Berlin: Springer-Verlag, 1988. ISBN 0387188991.

SCHENDEL, U. *Sparse matrices : numerical aspects with applications for scientists and engineers*. Chichester : New York: Ellis Horwood ; Halsted Press, 1989. ISBN 074580635X.

SCIAVICCO, L.; SICILIANO, B. *Modeling and Control of Robot Manipulators*. [S.l.]: McGraw-Hill, 1996. (Electrical and Computer Engineering). ISBN 0070572178.

SELIG, J. M. *Geometrical Methods in Robotics*. [S.l.]: Springer-Verlag, 1996.

SEMPLE, J. G.; KNEEBONE, G. T. *Algebraic Projective Geometry*. [S.l.: s.n.], 1952.

SESHU, S.; REED, M. B. *Linear Graphs and Electrical Networks*. Reading: Addison-Wesley, 1961.

SICILIANO, B. Closed-loop inverse kinematics algorithms for redundant spacecraft/manipulator systems. In: Werner, Robert; O'Conner, L. (Ed.). *Proceedings of the 1993 IEEE International Conference on Robotics and Automation: Volume 3*. Atlanta, GE: IEEE Computer Society Press, 1993. p. 95–100. ISBN 0-8186-3450-2.

SPONG, M. W.; VIDYASAGAR, M. *Robot Dynamics and Control*. New York: John Wiley and Sons, 1989.

STEWART, D. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, v. 180, n. 15, p. 371–386, 1966.

TAHMASEBI, F. *Kinematic Synthesis and Analysis of a Novel Class of Six-DOF Parallel Minimanipulators*. Tese (Doutorado) — The University of Maryland, 1992.

TISCHLER, C. *Alternative Structures for Robot Hands*. Tese (Doutorado) — The University of Melbourne, Melbourne, 1995.

TISCHLER, C. R. *Alternative Structures for Robot Hands*. Tese (Ph.D. Dissertation) — University of Melbourne, 1995.

TISCHLER, C. R.; HUNT, K. H.; SAMUEL, A. E. On optimizing the kinematic geometry of a dextrous robot finger. *International Journal of Robotics Research*, v. 17, p. 1055–1067, 1998.

TISCHLER, C. R. et al. Screw geometry and Ball's inertia. In: LIPKIN, H.; DUFFY, J. (Ed.). *Ball 2000 Conference*. Trinity College: University of Cambridge CDROM, 2000. p. 1–14:Ball2000–28.pdf.

TISCHLER, C. R.; SAMUEL, A. E.; HUNT, K. H. Kinematic chains for robot hands: Part 1 orderly number-synthesis. *Mechanism and Machine Theory*, v. 30, n. 8, p. 1193–1215, 1995.

TISCHLER, C. R.; SAMUEL, A. E.; HUNT, K. H. Kinematic chains for robot hands: Part 2 kinematic constraints, classification, connectivity, and actuation. *Mechanism and Machine Theory*, v. 30, n. 8, p. 1217–1239, 1995.

TISCHLER, C. R.; SAMUEL, A. E.; HUNT, K. H. Dextrous robot fingers with desirable kinematic forms. *International Journal of Robotics Research*, v. 17, p. 996–1012, 1998.

TREVELYAN, J. P. *Robots for shearing sheep : shear magic*. Oxford ; New York: Oxford University Press,, 1992. ISBN 0198562527.

TREVELYAN, J. P.; KOVESI, P. D.; ONG, M. C. H. Motion control for a sheep shearing robot. In: *Robotics Research*. [S.l.: s.n.], 1983. p. 175–190.

TSAI, L.-W. *Robot Analysis: the Mechanics of serial and parallel manipulators*. New York: John Wiley & Sons, 1999. ISBN 0-471-32593-7.

TSAI, L.-W. *Mechanism Design: Enumeration of Kinematic Structures According to Function*. Boca Raton: CRC Press, 2001. ISBN 0-8493-0901-8.

VALDIERO, A. C. et al. Cálculo e análise do jacobiano de um manipulador paralelo de 3 graus de liberdade baseado na teoria dos helicóides. In: *Proceedings of the Brazilian Congress of Mechanical Engineering -IX COBEM*. Uberlandia: Brazilian Society of Mechanical Sciences, 2001.

WALDRON, K. J.; WANG, S. L.; BOLIN, S. J. A study of the Jacobian matrix of serial manipulators. *Trans. ASME, Journal of Mechanisms, Transmissions and Automation in Design*, v. 107, p. 230–238, 1985.

WANG, S.-L.; WALDRON, K. J. A study of the singular configurations of serial manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, v. 109, p. 14–20, 1987.

WOO, L.; FREUDENSTEIN, F. Application of line geometry to theoretical kinematics and the kinematic analysis of mechanical systems. *Journal of Mechanisms*, v. 5, p. 417–460, 1970.

WU, W.; HUANG, Y. The direct kinematic solution of the planar Stewart platform with coplanar ground points. *J. Comput. Math.*, v. 14, n. 3, p. 263–272, 1996.

ZABLONSKI, K. I.; BELOKONEV, I. M.; TCHEKIN, B. *Teoria de Mecanismos e Máquinas*. Kiev, Ucrânia: Edições Ensino Superior, 1989. Em russo.

ZHANG, C.-d.; SONG, S.-M. Forward kinematics of a class of parallel (stewart) platforms with closed-form solutions. *J. Rob. Syst.*, v. 9, n. 1, p. 93–112, 1992.

ZOMAYA, A. Y.; SMITH, H.; OLARIU, S. Computing robot Jacobian on meshes with multiple buses. *Microprocessors and Microsystems*, v. 23, p. 309–324, 1999.