MODEL-BASED CONTROL OF FAST PARALLEL ROBOTS: a Global Approach in Operational Space

THÈSE N° 1228 (1994)

PRÉSENTÉE AU DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

PHILIPPE GUGLIELMETTI
Ingénieur informaticien diplômé EPFL
originaire de Morbio Superiore (TI)

acceptée sur proposition du jury:

Prof. R. Longchamp, rapporteur
Dr I. Vaclavic, corapporteur
Prof. J.-P. Merlet, corapporteur
Prof. R. Clavel, corapporteur

Lausanne, EPFL
1994

à Laura et Annick

Remerciements.

Il s'agit ici de profiter de la seule page où le formalisme n'est pas de rigueur pour saluer, embrasser, remercier un maximum de gens cher au coeur de l'auteur et/ou ayant peu ou prou contribué de quelque façon que ce soit à la gestation relativement prolongée de cet opuscule en minimisant le risque d'en vexer par omission, en évitant absolument de paraître flagorneur à l'encontre d'aucuns tout en saisissant l'occasion de pondre dans la langue d'Achille Talon une phrase dont la transposition dans celle de Woody Allen restera quelque temps encore hors de la portée de son auteur.

C'est à cause de Dominique Bonvin que le reste de ce document n'est pas de la même veine, mais c'est aussi en partie grâce à lui si les choses sont exprimées plus simplement. Qu'il soit ici remercié avec le Prof. Roland Longchamp dont la patience et le soutien infaillible m'ont accompagné tout au long de mon long mais O combien profitable séjour à l'Institut d'Automatique. Ma gratitude va également à Reymond Clavel, Jean-Pierre Merlet et Ivan Vaclavik pour leur sérieux lors de la correction du manuscrit, laquelle a permis d'éliminer nombre d'erreurs de la version finale.

De nombreuses idées exposées dans cette thèse trouvent leur source dans les passionnantes discussions avec Friedhelm Altpeter, Thomas Baumann, Olivier Chételat, Alain Codourey, Fathi Ghorbel, Karol Miller, Piotr Myszkorowski, Dan Necsulescu, Laurent Rey, Johannes Steinmetz, Boris Stevens et Peter Vischer. Merci à tous du fond du coeur! Mention spéciale àPascal Arbenz, Mahmoud Partovi, Patrick Thielen et Can Tuncelli pour leurs indispensables contributions à coups d'OCCAM et de [ctrl]-[alt]-[del], de fer à souder ou de fraiseuse. Vivent Big Mac, Frame Maker, Canvas, Mathematica, Matlab, Denis Gillet et Christophe Salzmann!

N'oublions pas les "pères spirituels" rencontrés au long de ces (nombreuses ?) années à l'IA : Pascal Hulliger, Jean-Daniel Marcuard, Mahmoud Partovi, Marc-André Eggiman sans oublier le plus souriant des barreurs de 470: Can Tuncelli. Marie-Claire et Monsieur Roch sont du fond du coeur remerciés d'exister.

Enfin, toute ma gratitude a ma famille pour m'avoir non seulement supporté si longtemps, mais nourri autant d'encouragements que de vitamines. Et si les progrès de ma fille Laura pendant cette dernière année m'ont définitivement découragé de faire de la robotique, ce travail lui est cependant dédié ainsi qu'à ma chère Annick pour m'avoir confirmé que la liberté ne se mesure pas en degrés.

Version Abrégée.

Une approche globale au problème de la commande des robots parallèles rapides est proposée dans ce travail. Les différences fondamentales entre les robots série et parallèles sont mises en évidence. Un formalisme inspiré de celui de Denavit-Hartenberg permet de paramétriser la géométrie d'un robot parallèle quelconque en le considérant comme deux arbres articulés connectés par six liaisons simples.

Puis, il est montré que la modélisation de la cinématique et de la dynamique des robots parallèles est grandement facilitée si on choisit de représenter l'état du robot simultanément par les variables décrivant la position des articulations motorisées et par les variables spécifiant la position de l'organe terminal du robot dans l'espace opérationnel. Une forme générale du modèle de la dynamique inverse dit "dans les deux espaces" découle de cette approche.

Un algorithme basé sur l'approche Newton-Euler est développé pour le calcul rapide et efficace du modèle de la dynamique inverse dans les deux espaces. Il est montré que sa complexité est comparable à celle de l'algorithme standard pour les robots série. Une formulation par la mécanique lagrangienne permet l'analyse du modèle dans les deux espaces. Il est montré que ses propriétés correspondent à celles bien connues des robots série à une importante différence près: la matrice Jacobienne des robots parallèles n'étant pas bornée, ces manipulateurs n'offrent des performances élevées que dans un sous-espace de leur espace de travail qui doit être soigneusement défini.

Différentes stratégies pour le réglage en poursuite de trajectoire des robots rapides sont ensuite examinées. Les avantages des méthodes basées sur la connaissance du modèle dynamique et une contre réaction robuste sur les variables opérationnelles sont décrits. Il est montré que pour les robots parallèles, une telle commande nécessite une puissance de calcul moindre lorsque la contre-réaction est réalisée dans l'espace opérationnel, c'est-à-dire lorsque le réglage s'effectue sur les coordonnées décrivant l'état de l'organe terminal du robot. Outre les avantages déjà énumérés dans la littérature, une telle structure peut être implantée de façon très efficace sur un contrôleur multiprocesseurs en exploitant la nature intrinsèquement parallèle des algorithmes impliqués ainsi que leur mise en pipeline.

Finalement, l'approche proposée est appliquée au robot parallèle Delta. Une démarche systématique permet d'obtenir les modèles cinématiques et dynamiques de ce robot sous une forme extrêmement compacte. L'analyse de la matrice Jacobienne ainsi que quelques résultats de simulation révèlent les points forts et les limitations de ce manipulateur. L'implantation d'une commande basée sur le modèle du Delta sur un contrôleur doté de quatre Transputers est décrite. Quelques résultats expérimentaux obtenus sur un Delta doté d'une transmission à courroie crantée sont présentés et discutés.

Abstract.

A global approach to the problem of model-based control of fast parallel robots is proposed in this work. Fundamental differences between the well-known serial arms and parallel manipulators are first explained. A formalism inspired form Denavit-Hartenberg's makes it possible to parametrize any parallel manipulator by handling it as two tree robots connected through six standard links.

Then, it is shown that kinematics and dynamics modelling is greatly simplified when the robot's state is represented both by the variables associated to the actuated joints and the variables specifying the end-effector's position in operational space. The inverse dynamics model of any parallel manipulator can then be put under a standard form called "in the two spaces".

A Newton-Euler based algorithm is proposed for the real-time computation of the model in the two spaces Its complexity is shown not to be much larger than for serial arms. Through Lagrangian mechanics, the model in the two spaces allows analysis of the robot's dynamics properties, such as passivity. These properties are shown to be equivalent to those of serial arms, except that parallel robots offer good performances only in a restricted workspace in which their Jacobian matrix remains bounded.

Various control strategies for the trajectory tracking problem for fast robots are then examined. The advantages of model-based approaches combined with robust feedback laws in operational space are described. It is shown that a control loop in operational space requires less computations than in joint space for a parallel robot. Moreover, such a scheme can be very efficiently be implemented on a multiprocessor control unit that exploits the intrinsically parallel and pipeline structure of the required algorithms.

Finally, the proposed approach is applied to the Delta parallel manipulator. A systematic approach leads to the kinematics and dynamics models of this robot, which are expressed under a very compact form. The analysis of the Delta's Jacobian matrix as well as some simulation results reveal the advantages and weak points of this manipulator. The implementation of a model-based control law for the Delta on a control unit with four Transputers is described. Some results obtained on a Delta with a crank belt reduction are presented and discussed.

Table of Contents.

Remerciements. 5

Version Abrégée. 7

Abstract. 9

Chapter I : Objectives, Motivations and Preliminaries. 17

- **I.1 Thesis.** 17
- **I.2 Report's Outline.** 17
- **I.3 Robotics and Control.** 19
  - I.3.1 Preliminary Definitions.
  - I.3.2 Robotics: a Multidisciplinary Technology.
- **I.4 Serial Arms and their Limitations.** 20
- **I.5 Parallel manipulators.** 21
  - I.5.1 Closed Chains.
  - I.5.2 Some History.
  - I.5.3 The Delta Robot Approach.
  - I.5.4 State of the Art and Research Topics.
- **I.6 .Structure of a Control Unit.** 23
  - I.6.1 Operational Space Control versus Joint Space Control.
  - I.6.2 Control of High-Performance Manipulators.

Chapter II : Manipulator Kinematics. 25

II.1 Introduction. 25
II.2 Degrees of Freedom of Rigid Bodies. 26
II.2.1 Bodies and Joints.
II.3 Mobility of Articulated Structures. 27
II.3.1 Open-Chain Structure.
II.3.2 Closed-Chain Structure.
II.3.3 Configuration Space and Joint Space.
II.3.4 Operational Space.
II.3.5 Non–Holonomic Robots.
II.3.6 Redundant Robots.
II.4 Sensors and Actuators. 29
II.4.1 Actuators and Position Sensors.
II.4.2 Velocity Sensors.
II.4.3 Accelerometers.
II.5 Geometry of Manipulators with Open Chains Only. 31
II.5.1 Serial Arms.
II.5.2 Tree–Robots.
II.5.3 The Denavit–Hartenberg Parametrization.
II.6 Geometry of Manipulators with Closed Chains. 34
II.6.1 Parallelograms and 5–Bars Mechanisms.
II.6.2 Parallel Robots.
II.6.3 Extension to Cooperative Robots.
II.6.4 Limitations of the Denavit–Hartenberg Formalism.
II.6.5 Natural Coordinates.
II.7 A New Formalism for Parallel Manipulators. 36
II.7.1 Specificities of Parallel Manipulators.
II.7.2 A Wide Class of Parallel Manipulators.
II.7.3 The Double Tree Model.
II.7.4 Equivalent Structure.
II.7.5 Real and Abstract Links.
II.7.6 Geometric Parametrization.
II.8 Coordinates Transformations. 41
II.8.1 Forward Kinematics.
II.8.2 Inverse Kinematics.

II.9 Differential Kinematics: the Jacobian and its Inverse. 44
    II.9.1 The Jacobian Matrix of Open Chain Mechanisms.
    II.9.2 Singularities in the Jacobian.
    II.9.3 The Inverse Jacobian Matrix of Open Chain Mechanisms.
    II.9.4 The Inverse Jacobian of Parallel Manipulators.
    II.9.5 Inverse Singularities.
    II.9.6 Jacobian of Parallel Robots.
    II.9.7 Extended Jacobian Matrix and Absolute Accuracy.
    II.9.8 Acceleration Kinematics.
II.10 Desired Task and Generation of Reference Signals. 50
    II.10.1 Task Description.
    II.10.2 The Profile Function.
    II.10.3 Conversion to Joint Space.
II.11 Conclusions. 52

Chapter III : Manipulator Dynamics. 53

III.1 Introduction. 53
III.2 Rigid Body Dynamics: an Overview. 54
    III.2.1 Newton–Euler for Control.
    III.2.2 Lagrange for Analysis and Control.
    III.2.3 Modern Theory and Simulation.
III.3 Virtual Work Principle. 56
    III.3.1 Application to Static Stiffness Determination.
III.4 Lagrange Method. 58
    III.4.1 A Very General Approach.
    III.4.2 Physical Parameters.
III.5 Open Chain Manipulators. 60
    III.5.1 Inverse Dynamics Model.
    III.5.2 Formulation of the Model in Operational Space.
III.6 Parallel Manipulators. 62
    III.6.1 Relation between the Lagrange Multipliers and the Jacobian Matrix.
    III.6.2 Inverse Dynamic Model in the Two Spaces.
III.7 Efficient Computation of the Inverse Dynamics Model. 64
    III.7.1 Intuitive Justification.
    III.7.2 The Newton–Euler Algorithm for Open–Chain Manipulators.
    III.7.3 Extension.
    III.7.4 A New Algorithm for Parallel Manipulators.
    III.7.5 Complexity, Optimization and Structure of the Algorithm.

III.8 A General Framework: the Operational Space Formulation. 72
    III.8.1 Projection of the Model in the Two Spaces onto Operational Space only.
    III.8.2 Properties.
    III.8.3 Is Passivity a Local Property?
    III.8.4 A Critical Discussion on the Inertia of Manipulators.
    III.8.5 A Note on Discrete–Time Models.
III.9 Conclusions. 78

Chapter IV : Control. 79

IV.1 Introduction. 79
IV.2 Classical Control Strategies.. 80
    IV.2.1 Dynamics of Robots with Transmissions.
    IV.2.2 Decoupled Linear P(I)D Controllers.
    IV.2.3 The Model-Based versus Robust Control tradeoff.
IV.3 Model-Based Control. 87
    IV.3.1 Feedback Linearization.
    IV.3.2 Closing the Loop : the Computed-Torque Scheme.
    IV.3.3 Computation of the Inverse Dynamics.
    IV.3.4 Model-Based Control in Operational Space.
    IV.3.5 Advantages and Drawbacks.
    IV.3.6 Reported Experimental Results.
IV.4 Adaptive Control. 94
    IV.4.1 Adaptive Inverse Dynamics.
    IV.4.2 Passivity-Based Control Methods.
    IV.4.3 Simulation Results and Open Problems.
    IV.4.4 Experimental Results and Conclusion.
IV.5 Robust Control. 96
    IV.5.1 The Linear Multivariable Approach.
    IV.5.2 Variable Structure Controllers.
    IV.5.3 The Passivity Approach.
    IV.5.4 A Unified Point of View Thanks to Lyapunov.
    IV.5.5 Limitations of High Gain Feedback Control Laws.
IV.6 A Control Strategy for Fast Parallel Manipulators. 100
    IV.6.1 Applications of Parallel Manipulators and Associated Control Schemes.
    IV.6.2 Simplified Models for Fast Manipulators.
    IV.6.3 Reported Experimental Results.
    IV.6.4 Joint Space Versus Task Space Control Schemes.

IV.7 Implementation Considerations. 104
    IV.7.1 Previous Work on Controllers for Serial Arms.
    IV.7.2 Parallel Controllers for Parallel Robots.
    IV.7.3 Control Unit Design.
IV.8 Conclusions. 111

Chapter V : Application to the Delta Robot. 113

V.1 Introduction. 113
V.2 Experimental Setup. 114
    V.2.1 The Delta Manipulator.
    V.2.2 Geometrical Description.
    V.2.3 The Delta-IA Robot.
V.3 Kinematics. 119
    V.3.1 Preliminary Definitions.
    V.3.2 Inverse Kinematics.
    V.3.3 Inverse Jacobian.
    V.3.4 Forward Kinematics and State Observer.
    V.3.5 Inverse Jacobian's Time Derivative and the Acceleration Mapper.
V.4 Singularities and Definition of the Usable Workspace. 125
    V.4.1 Singularities and Inverse Singularities.
    V.4.2 Defining a Usable Workspace.
V.5 Dynamics. 129
    V.5.1 Modelling Gravity Effects.
    V.5.2 The Simplified Model.
    V.5.3 The Complete Model in the Two Spaces.
    V.5.4 Complete Versus Simplified Model: A Comparison.
    V.5.5 Implementation Issues.
V.6 Simulations, Experimental Results and Analysis. 135
    V.6.1 A Simple Test Path.
    V.6.2 A Simple Pick & Place Task.
    V.6.3 Identification of Actuator and Transmission Dynamics.
    V.6.4 Applicability of the Proposed Control Scheme.
V.7 Conclusions. 143

Chapter VI : Conclusions. 145

Appendix I : Notations. 149

Appendix II : List of Figures. 155

Appendix III : List of Tables. 157

Appendix IV : References. 159

Curriculum Vitae. 169

Chapter I

Objectives, Motivations and Preliminaries.

I.1 Thesis.

The objective of this work is to provide a complete approach to the problems of designing a trajectory tracking controller for high-speed parallel robots and to efficiently implement it. Many different topics are discussed, but the underlying thesis can be stated in a few words as follows:

A wide class of parallel robots can be modeled as two partial tree structures connected through simple links. The corresponding dynamics model is called "in the two spaces" since it uses the robot's state simultaneously in joint space and in operational space. This formulation makes it possible to find passive mappings in the dynamics of parallel manipulators. A robust, model-based control law can therefore be designed in operational space to achieve high-performance trajectory tracking. The intrinsic parallel-pipeline structure of the corresponding algorithms can be exploited on a multiprocessor control unit.

I.2 Report's Outline.

Objectives, Motivations and Preliminaries.

The remaining of this chapter is intended as a presentation of the objectives of this work and the motivations behind it. In Section I.3, frequently used words are defined and robotics is presented as a union of many different technologies, where dynamics is the glue that allows the analysis and the design of all the elements in a robot's feedback loop.

Most of today's industrial manipulators, such as those used in car factories, are the so-called *serial arms*. Section I.4 explains their limitations when fast, high-performance operation is required. A newer family of manipulators called *parallel robots* allow better dynamical performance. In Section I.5, their history is sketched and it is shown that their structure poses additional and different problems than that of serial arms.

The structure of a robot's control unit is then discussed in Section I.6. Since the controller can either be designed to directly act on the joints' motors or virtually act on the robot's end-effector coordinates, different kinematics transformations must be added to the control law itself. This intimate relationship between dynamics and geometry is exploited throughout this work.

Kinematics.

In Chapter II, after some definitions, a new classification and modeling method for manipulators with closed chains is proposed. It is based on the idea that any parallel robot can be considered as two tree robots connected through simple links with free joints at both edges.

Then, the kinematics of serial arms and parallel robots are discussed and their fundamental differences are shown: forward kinematics transformations are very simple for open chain mechanisms, but they do not exist in closed-form for general manipulators with closed chains. Conversely, inverse *kinematic transformations are simpler for parallel manipulators than those for serial arms.*

Dynamics.

In Chapter III, Lagrangian dynamics is applied to parallel manipulators. It is shown that the widely used "Lagrange with multipliers" approach contains a hidden computation of the Jacobian matrix. When the Jacobian is explicitly introduced, an inverse dynamics model "in the two spaces" can be obtained for any parallel robot. A new algorithm to compute this model systematically and efficiently is derived, based on the well-known Newton-Euler algorithm for open chain robots. It is shown that the complexity of the new algorithm is comparable with the latter.

The model "in the two spaces" can be formulated using operational space coordinates only, where it becomes a generalization of the model in operational space of serial arms. Analysis is then carried out with this very general formulation. It is proved for the first time that passive mappings exist in the dynamics of parallel manipulators as for serial arms. However, this only holds in a restricted workspace of the robot where singularities of the inverse Jacobian are excluded. In this respect, some critical remarks about the alleged higher stiffness and lower inertia of parallel manipulators are also made.

Control.

The control problem is covered in Chapter IV. The application of existing approaches to parallel robots is first discussed. It is shown that most of them use an explicit model of the robot for feedforward control and passivity as a tool to ensure some robustness in the feedback control. The advantages of operational space control schemes for parallel robots are presented. Finally, a controller which uses a model in the two spaces and a robust feedback law in operational space is proposed. Among other advantages, it is shown that the corresponding algorithm requires less computations than the equivalent scheme in joint space.

Implementation issues are then discussed. A multiprocessor control unit is shown to be well suited to compute the inverse dynamics model and the corresponding kinematic transformations. The intrinsical parallel structure of the algorithms involved is then fully exploited.

Application to the Delta Robot

The proposed modeling approach and control strategies are applied to the Delta parallel robot in Chapter V. The obtained algorithms for kinematics and dynamics computations are detailed and their implementation on a multiprocessor control unit studied. A numerical analysis of the robot's Jacobian matrix is performed, which shows the importance of defining the robot's usable workspace properly. A Lagrange-based inverse dynamics model "in the two spaces" is developed and analyzed. Finally, some simulation and experimental results lead to a critical discussion on the Delta's performances and limitations.

I.3 Robotics and Control.

I.3.1 Preliminary Definitions.

A *manipulator* is a mechanism able to move a *gripper* or a *tool* in a certain *workspace* so that a certain *task* can be automatically performed. The public and end-users normally uses the word "*robot*" to refer to the whole system that actually executes the task, i.e. the manipulator with its *actuators* and their *power amplifiers*, the *sensors* and the *control unit*, where a controller is implemented.

In Fig. 1, the robot is seen from the controller as a controlled plant in the natural sense, where the manipulator refers only to a part thereof.

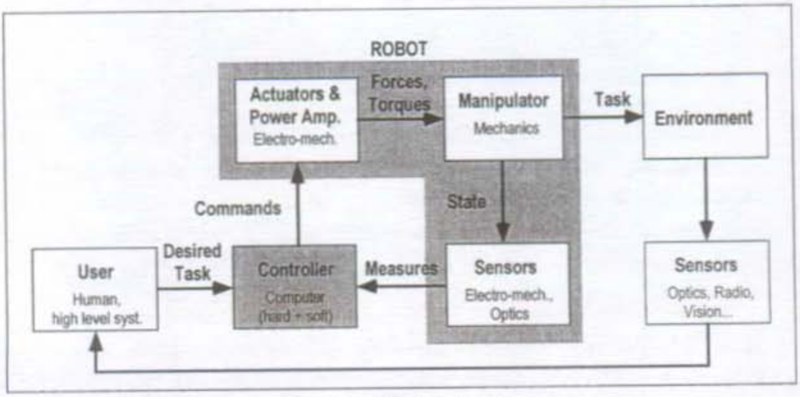


Fig.1 : A Complete Robotized Plant.

The controller provides the power amplifiers with command signals that are computed from the sensors' measurements. This closed loop allows the robot to execute any task defined by the user. The appearance of modem real-time computers and software techniques allows to build very complex and powerful control units that can greatly enhance the robot's performance.

I.3.2 Robotics: a Multidisciplinary Technology.

A robot integrates many different technologies such as electro-mechanics, hydraulics or pneumatics, mechanical construction, materials sciences, optics and signal processing. If one considers the whole robotized plant shown in Fig. 1, other technologies such as artificial intelligence or discrete event systems may even be considered when dealing with the outer loop and its higher-level control system.

From the implementation point of view, the control unit's hardware and software have to be developed according to the structure of the control law. This requires expertise in algorithmic, real-time and possibly multiprocessing computer systems.

In this work, we consider all these problems from the point of view of dynamics. Since the robot's tasks and performances are specified using time and/or frequencies, dynamics provides a uniform framework for the robot's design and analysis and for the synthesis of a control law. The main problem is that modem manipulators exhibit nonlinear behaviors that can not be treated using classical control theory.

The manipulator's nonlinearity is a direct consequence of its geometrical structure and useful properties of its dynamics can therefore be derived from a purely geometrical analysis. Hence, the kinematics and dynamics must be discussed before the control problem and the controller design are addressed.

I.4 Serial Arms and their Limitations.

Today, the most widely used industrial manipulators are the so-called *"serial arms"*. They are made of an *open kinematic chain* of *bodies* connected by *joints*. Linear and/or rotary motors are mounted on the chain and drive each joint through a *reduction* which is typically a *gearbox*. The indisputable advantages of serial arms are a simple structure, a large workspace compared to the robot's size, and a complete theory about their design, control and operation.

However, their dynamical performance is rather poor since their structure often has to carry the actuators, at the expense of increasing their inertia. Moreover, each link has to bear flexion stresses that introduce flexibility in the joints or, even worse, in the links themselves. As a consequence, the first resonance frequency of a typical serial arm like the PUMA 560 is not higher than 10Hz [Leah89].

Researchers have therefore tried to explicitly consider flexibility in the dynamics and to develop controllers that allow the closed-loop bandwidth to exceed the resonance frequency of the arm. However, this approach requires more sensors and a much higher computational burden for the controller. Even if very interesting theoretical work has been done in this field, one point becomes clear : serial structures are not well suited for fast manipulators.

I.5 Parallel manipulators.

I.5.1 Closed Chains.

Introducing *closed chains* in the manipulator structure has very positive effects on its stiffness because most of the bodies bear much smaller flexion stress. Since the structure itself is used as a transmission, actuators can be placed closer to the base or even be fixed on it.

I.5.2 Some History.

Mathematicians such as Cauchy, Lebesgue or Bricard already studied structures with closed chains more than one century ago. However, their use for manipulators could not be imagined before 1938, when Pollard [Poll38] took out a patent for a car painting robot, but the technology did not allow him to control his invention at that time. Then, Mc Gough [Goug62] proposed in 1947 the parallel structure with six prismatic actuators which is now erroneously called *'Stewart platform'*. Actually, Stewart [Stew65] proposed a quite different design for a mobile platform to carry flight simulators' cockpits.

Using this type of mechanisms for industrial manipulators has been considered by a few researchers, leading only recently to actual parallel manipulators. Extensive directories of such robots may be found in [Clav91] and [Merl90]. The latter reference is the most complete textbook on parallel manipulators, if not the only existing one.

Before the development of the Delta robot, parallel structures were used for applications that require strength and accuracy like *assembly tasks*, but not really high speed. Most of these machines were rather heavy, some even used hydraulic actuators.

I.5.3 The Delta Robot Approach.

A new generation of parallel robots appeared in 1985 with the development of the Delta robot at the Institut de Microtechnique (IMT) of EPFL by Clavel [Clav85], [Clav88], [Clav91]. His aim was to build a very fast 3-degrees-of-freedom robot for *pick-and-place* tasks. This successfully resulted in one of the fastest industrial robot currently available (see Fig. 2). The Delta is now manufactured and exploited mostly in the packaging industry.

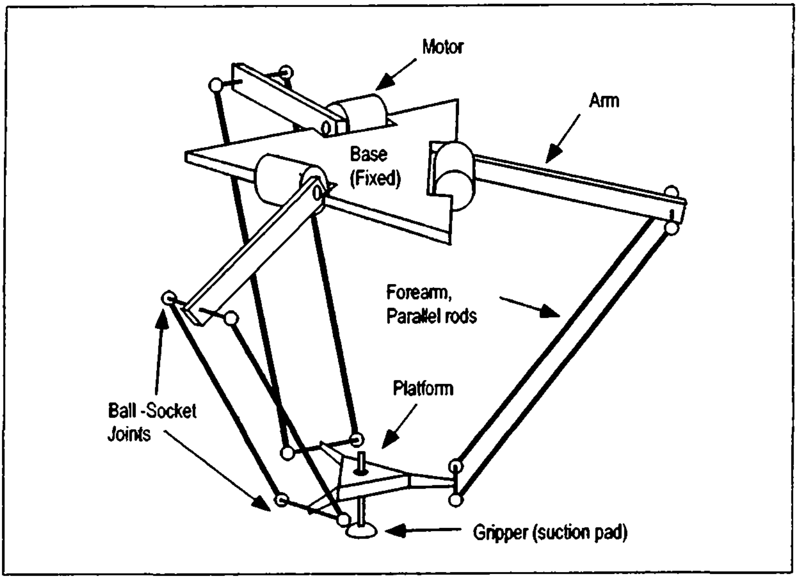


Fig.2 : The Delta Robot.

I.5.4 State of the Art and Research Topics.

Since then, other fast manipulators were built, that are inspired from the Delta structure, for example the *Hexa* [Pier91], [Pier92], [Pier92] or the *Speed-R-Man* [Rebo90],[Rebo91].

All these new parallel robots outperform in speed any serial robot with a similar workspace. The fundamental reason is that *their natural frequency is quite higher. The main drawback of parallel* manipulators is the relatively small volume of their workspace compared to serial arms of corresponding sizes.

Up to now, the robotics and control scientific communities have considered parallel manipulators as rather original and peculiar mechanisms. Most of the research in robotics does not handle the case of parallel robots. In fact, very few of the available results in the field of robot control can be applied to them.

Most of the research has until now focussed on kinematics and static considerations about each special structure. The few attempts to provide a systematic approach to the dynamics modelling, analysis and control of parallel manipulators gave extremely complicated results that are useless for the engineer.

I.6 .Structure of a Control Unit.

I.6.1 Operational Space Control versus Joint Space Control.

The relationship between kinematics and dynamics can be understood intuitively by looking closer at the functions included in a control unit.

- First, the information given by the user about the task to perform and the measurements made on the robot and/or the commands sent to it are not directly compatible. On one hand, the desired task is specified in the *operational space* where the position and orientation of the end-effector is directly described relatively to a *reference frame* attached to the fixed *base* of the robot. Any user-defined *coordinates system* may be used in this respect. On the other hand, the sensors give information about the robot's *joints* and the actuators exert forces or torques on them. Hence, the actual inputs and outputs of the manipulator are specified in a *joint space* that reflects the robot's *configuration* measured by the sensors. The control unit must do some geometric conversions before comparing the user's objectives and the measured data. Since they also involve speeds and accelerations, they are rather referred to as *kinematics transformations*.
- Second, the controller will compute the commands to send based on compatible values of the *reference trajectory* and measures. The design of this part of the control unit relies on dynamics.

Hence, two different structures of controllers are possible according to the direction of the conversions, as shown in Fig. 3.

Operational space control may seem more involved than joint space control since it requires three blocks instead of two. However, an important contribution of this work is to show that *parallel* robots are more easily described in operational space, leading to much simpler operational space controllers

I.6.2 Control of High-Performance Manipulators.

Many of today's industrial robots are operated at low speeds and the *reduction* ratio of their motors is so high that nonlinear dynamics effects can be neglected and each degree-of-freedom considered as independent. The structure of the corresponding control unit is then a two-level hierarchy where multiple independent, single input-single output (SISO) linear controllers at the lower level each control one of the robot's actuators. At the upper level, reference signals and optional feedforward information is computed and fed to each controller.

Enhancing the dynamical performance of manipulators is equivalent to brodening the bandwidth of the controlled robot, which requires to improve each of the elements in the loop. Nowadays, sensors and electric drives meet the requirements of the modern robotics market. New articulated structures and composite materials allow to build much stiffer manipulators. Today, the challenge for a robot's designer is to improve the controllers.

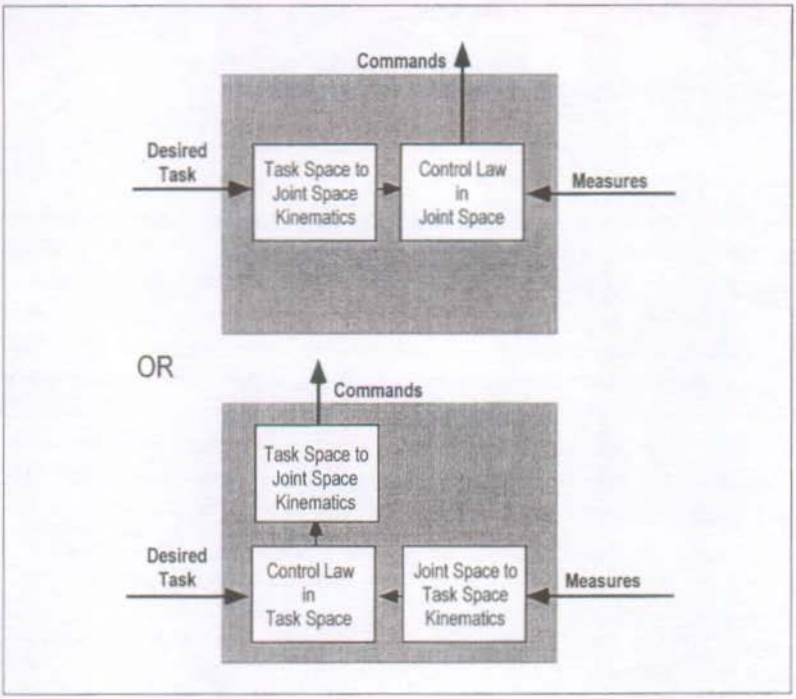

Commands
Desired Task
Task Space to Joint Space Kinematics
Control Law in Joint Space
Measures
OR
Commands
Task Space to Joint Space Kinematics
Desired Task
Control Law in Task Space
Joint Space to Task Space Kinematics
Measures

Fig.3 : Joint Space Control or Operational Space Control

Effects such as couplings, Coriolis and centripetal forces that were neglected until now can no more be ignored when controlling a fast robot. Taking them into account requires new tools from the control theory and more computing power in the control unit. Even the logical and hardware structures of the control unit must be changed since a hierarchical structure is not adequate when couplings are considered.

However, the fundamental source of these nonlinearities resides in the equations of the movement of each of the bodies composing the robot's mechanical structure. A good understanding of the manipulator's kinematics is therefore required before its dynamics can be analyzed and a proper controller synthetized.

Chapter II

Manipulator Kinematics.

II.1 Introduction.

In this chapter, an approach to manipulators' kinematics is presented that differs from the one given in most robotics textbooks such as [Spon89]. In some textbooks, parallel manipulators are considered as special cases and quickly handled as tree–robots with closure constraints, thus avoiding some difficult real problems [Domb88]. The aim here is to emphasize the fundamental differences between the well–known serial robots and the less studied parallel structures. These differences have important consequences on dynamics, control and even on the control unit's design.

Rigorous definitions and basic concepts of robotics that will be used throughout this thesis are presented. All the corresponding symbols and notations are listed in Appendix I. The notion of degree–of–freedom is defined in Section II.2 and the mobility of an articulated structure in Section II.3. Section II.4 shortly reviews the nature and location of the manipulator's sensors and actuators.

Manipulator structures with open kinematic chains are described in Section II.5 and their geometric parametrization based on Denavit–Hartenberg parameters are discussed. In Section II.6, closed kinematic chains are introduced and the limitations of the existing parametrizations are discussed.

A new formalism for the description of parallel manipulators is introduced in Section II.7. Within this methodology, a very wide class of parallel manipulators is described as two tree structures connected to each other through six simple links. This allows to classify such mechanisms and provides a systematic way to obtain their kinematics and dynamics models.

Section II.8 compares the form of the kinematics transformations performed by serial and parallel mechanisms. The fundamental result of this comparison is that serial arms are more easily described by forward kinematics relations while parallel robots have a simpler formulation using inverse kinematics.

This difference entails important consequences on the differential kinematics models that can be obtained for serial or parallel manipulators. They are dealt with in Section II.9, where an extensive discussion of the Jacobian matrix of manipulators is made.

A first application of kinematics is considered in Section II.10. The use of inverse kinematics to convert reference signals from operational space to joint space is discussed. It is shown that the required computations exhibit a different structure for parallel and serial manipulators.

As a conclusion to this chapter, the fundamental differences between serial arms and parallel manipulators are summarized in a comprehensive table in Section II.11.

II.2 Degrees of Freedom of Rigid Bodies.

The *mechanical structure* of a manipulator is made up of *bodies* connected by *joints*. In this work, the bodies are considered as perfectly rigid since the robot's closed loop bandwidth is limited below the manipulator's resonance frequencies (see Chapter IV).

A single free rigid body has six *degrees of freedom (DOFs)* since it can be translated in three independent directions and rotated around three independent axes.

The *mobility* of a structure is defined as the number of independent variables that uniquely define the position and orientation of each of its elements in space.

II.2.1 Bodies and Joints.

A structure made of n_b connected bodies has *less* than $6n_b$ DOFs since *constraints* are introduced by the n_j *joints*. The ith joint in the structure is said to be a n_i–DOF joint if it allows n_i independent relative movements between the two bodies it connects.

A *prismatic joint* is a 1–DOF joint that allows a relative translation of the two connected bodies along a common axis and a *revolute joint* allows a relative rotation around the axis. In these cases a single *joint variable* is required to specify uniquely the relative position of the two bodies.

A joint driven by an actuator is an *actuated joint*. Only 1–DOF actuated joints exist in practice since motors for more complex joints are too expensive and not adequately efficient. Non–actuated joints are called *free joints*.

Any joint with $1 < n_i < 6$ can be considered as a combination of revolute and prismatic joints connecting imaginary punctual and massless bodies. However, only a subset of those joints is actually feasible from the mechanical engineering point of view and can be applied to manipulator's structures. Parallel robots contain *free joints* made up of *universal joints* that allow two rotations or *ball–socket joints* that allow three rotations.

II.3 Mobility of Articulated Structures.

II.3.1 Open-Chain Structure.

When no closed path exists along the bodies of the manipulator, it has an *open chain* structure. By convention, the n_b bodies include the fixed *base* and the structure therefore contains $n_j = n_b - 1$ joints with a total mobility of

$$n_q = \sum_{i}^{n_j} n_i . \tag{II.1}$$

Note that adding one body to such a structure implies adding exactly one joint to connect it, thus increasing the mobility of the whole by one.

II.3.2 Closed-Chain Structure.

A way to remove degrees of freedom in a structure is to add a joint without adding a body. A *closed chain* is therefore created. Joints in a closed chain are not independent: it is not possible to move a single joint in the loop without creating a displacement of at least another one because the loop satisfies a *closure constraint*. The mobility of a spatial structure is given by the *Grübler formula* [Merl90],[Clav91]

$$n_q = 6(n_b - n_j - 1) + \sum_{i}^{n_j} n_i , \tag{II.2}$$

where n_i is the mobility of the i^{th} joint in the closed chain.

However, this formula is valid only when the structure is in a generic configuration where no superabundant constraints place the system in an hyperstatic state. Moreover, note that Eq. II.2 gives the total mobility of the structure, including internal degrees–of–freedom that may allow a body to rotate about an axis without causing any displacement in the remaining of the structure.

II.3.3 Configuration Space and Joint Space.

For any given articulated structure, Eq. II.1 and Eq. II.2 imply the existence of a vector containing n_q independent variables that are sufficient to characterize in a unique way the position of all the bodies. This vector defines the robot's *configuration* within a *configuration space* $\mathcal{C} \subset \mathcal{R}^{n_q}$.

A manipulator should therefore have at least n_q *actuators* on independent joints to be controllable and at least n_q measures of independent joint variables to be observable. All the measured information about the manipulator's configuration is therefore collected in a vector q of n_q independent joint variables. q is defined in the *joint space* $\mathcal{A} \subset \mathcal{R}^{n_q}$.

Hence, the robot must be designed and operated in such a way that a one–to–one mapping $\mathcal{A} \leftrightarrow \mathcal{C}$ exists (see § II.4.1).

II.3.4 Operational Space.

As defined in § I.3.1, the user's goal is to have some task performed by the robot with its end–effector, not with the full manipulator structure. Since the end–effector is a body in space, its position can be specified by a vector p of $3 \le n_p \le 6$ independent variables, where n_p denotes its mobility. Any task can therefore be defined in the *operational space* $\mathcal{O} \subset \mathcal{R}^{n_p}$. A n_p–dimensional *operational coordinates system* also has to be defined. *Cartesian coordinates* are generally used in practice to specify the end–effector's position. The robot's *workspace* is defined as the projection of $\mathcal{O}$ onto the three–dimensional Cartesian space that contains all the possible end–effector positions.

If $n_p = 6$, the end–effector can take any *orientation* in the workspace. In practice, orientations are most often described by three–dimensional *Euler angles*, *direction cosines* or *roll, pitch and jaw angles* because of their intuitive geometrical meaning. However, these representations suffer from the existence of singularities and modulus problems that prevent to define a one–to–one $\mathcal{A} \leftrightarrow \mathcal{C}$ mapping. This is in fact a consequence of the non–integrability of the rotations in a three–dimensional space. *Four–dimensional Euler parameters*, *Olinde–Rodrigues parameters* or *quaternions* [Chou92] are equivalent ways to solve these difficulties but their less intuitive signification and the over–parametrization they introduce has restricted their use to the academic world.

If $n_p < 6$, the end–effector's position and orientation are no more independently controllable. While this may seem to limit the use of the robot, industrial applications have shown that many tasks only require only one DOF in orientation, or even none. In this case however, it is highly desirable that the remaining $6 - n_p$ orientation degrees–of–freedom of the end–effector are constant in the workspace.

II.3.5 Non–Holonomic Robots.

From Eq. II.1 and Eq. II.2, n_p may differ from n_q.

When $n_q < n_p$, the robot is said to be *non–holonomic*: in each point of the configuration space, additional constraints such as frictions forbid $n_p - n_q$ DOFs of the end–effector, which therefore remains controllable through the n_q actuators. However, the robot can not follow any trajectory in configuration space but may nevertheless reach any configuration after some *maneuvering*. Mathematically, non-integrable constraints appear between the system's velocities.

This is very common in mobile robotics where two independent actuators (propulsion and steering) may be used to move the robot in a three-dimensional space (position on a surface and orientation) thanks to the friction between the wheels and the ground. Non-holonomy is not considered further as such since no industrial manipulator takes advantage of this idea. However, the concept helps to understand the *inverse singularities* that play an important role for parallel manipulators (see § II.9.5).

II.3.6 Redundant Robots.

If $n_q > n_p$, the robot is said to be *redundant*. When the same end-effector position and orientation can be achieved through multiple (even infinite) configurations, a *kinematical redundancy* is reached. This property can be used to increase the robot's ability to avoid obstacles in the workspace and to optimize a performance criterion such as minimizing the energy spent during operation. When a closed kinematic loop exists, a redundant robot creates internal workless forces that can for example be used to modify the manipulator's stiffness on purpose [Adli90], [Adli91], [Kokk92]. This is a *dynamical redundancy*.

Since the modem approach to industrialization is to simplify the production process rather than to use expensive general tools, redundant manipulators are not common in industrial applications. Therefore, only general holonomic and non redundant manipulators with $n = n_q = n_p$ degrees of freedom are considered further in this work.

II.4 Sensors and Actuators.

II.4.1 Actuators and Position Sensors.

In an open chain mechanism, all the joints are actuated and q contains the measurements of all the associated relative positions. One-to-one $\mathcal{A} \leftrightarrow \mathcal{C}$ mappings are therefore as trivial as the identity.

In a structure with closed chains, only n_q joints are actuated and q can be defined by selecting n_q independent joint variables among the n_j joints. The mapping $\mathcal{A} \rightarrow \mathcal{C}$ can then be one-to-n_a, where n_a represents the number of possible configurations (called *assembly modes*) that are characterized by the same measured q vector. Using the knowledge of the initial configuration and considering that any q-trajectory of the robot in $\mathcal{A}$ should be continuous while the robot is moving, this mapping can be made one-to-one as long as singular positions (see § II.9.2) are avoided. Formally, however, this leads to very important and difficult problems that are not adressed in this work.

The ambiguity can also be removed by measuring extra joints. This solution has also other advantages (see § II.8.1) but requires additional (expensive) sensors as well as *sensory fusion* techniques to solve the coherency problem between the redundant measurements.

In practice however, the q vector always contains the measures of all actuated joint variables for different reasons:

- 1–DOF actuators and sensors are much simpler and cheaper than multi–degrees–of–freedom devices;
- *modern "intelligent" actuators often include an integrated sensor;*
- neglected elastic modes of the transmission or structure between actuators and sensors may lead the controlled system to instability. This is known in the robotics community as the *non–collocation* problem. See [Eppi93] for a good survey on this topic.

Physically, high–performance position sensors are generally optical incremental encoders. Electronic counters provide the control unit with a continuous–time (no conversion delay), digital estimation of the position. Since absolute errors of the sensors are extremely low, the main source of disturbances is the *quantification noise* which may be considered as white noise.

II.4.2 Velocity Sensors.

Most modern electric drives include a tachometer that gives an analog measurement of the motor's speed. In robotics, actuators often operate in a wide range of velocities, requiring expensive high–resolution A/D converters. Moreover, electromagnetic noise disturbs the signal at very low speeds and filtering results in unwanted negative phase shifts.

To overcome these difficulties, a common approach is to differentiate numerically the measured positions. However, the signal/quantification noise ratio becomes poor for low speeds where oversampling and filtering techniques are required [Hans88]. Some research has been done on velocity observers based on position measurements only but the quantification problem was not considered [Canu91], [Canu92].

A new idea for further research in this area is to inverse the time elapsed between two successive detections of increments. This gives better signal/noise ratios for low speeds but requires specific hardware to be developed.

II.4.3 Accelerometers.

Today's integrated accelerometers are much cheaper, lighter and more accurate than before. Acceleration measurements are required in some adaptive control schemes *(see Section IV.4). They may* also be used for the control of higher order dynamics of the robot such as joint elasticity. However, accelerometers are able to measure only linear accelerations of bodies in the robot's structure, not the joint's angular accelerations. Some nontrivial extra processing has therefore to be performed to convert the measured accelerations to the required coordinates system.

II.5 Geometry of Manipulators with Open Chains Only.

II.5.1 Serial Arms.

Most existing industrial robots are based on a single open chain structure with a fixed *base* at one edge and a single end–effector at the other. They are called *serial arms* or simply arms by anthropomorphism (see Fig. 4). Their structure can easily be described using *Denavit–Hartenberg parameters* that specify the geometry of each joint/body pair using four values that completely specify the position of the body relatively to the previous one in the chain. This parametrization is detailed in § II.5.3 below.

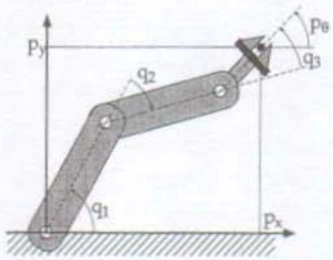


Fig.4 : Three Degrees of Freedom Serial Arm.

II.5.2 Tree–Robots.

Some robots have $n_e > 1$ end–effectors at the edge of multiple open chains. They are called *tree–robots* and can be considered as extended serial arms (see Fig. 5). This type of robot is presented here as an introduction to closed–chain mechanisms since the parametrization of parallel robots proposed in Section II.7 is based on the extended Denavit–Hartenberg parameters used for tree–robots.

II.5.3 The Denavit–Hartenberg Parametrization.

Denavit–Hartenberg parameters were proposed to describe the geometry of any serial arm using four parameters per joint, one of them being the joint variable and the three others being constants [Hart64]. This method was then extended to *tree–robots* by including only one more parameter per joint [Ait-92]. However, since the geometrical significance of this parameter is not immediate, most authors prefer to add two [Khal87], [Domb88]. The formalism from the latter reference is reviewed here.

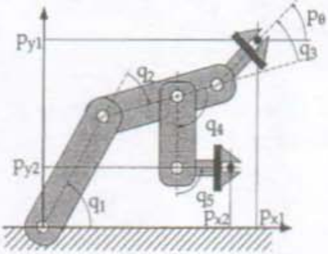


Fig.5 : "Three plus Two" DOFs Tree–Robot.

The n body/joint pairs in the structure are numbered from 1 to n so that the indexes are growing along every open chain from the base to the end–effector. By convention, the robot's base is considered as the 0th body. Moreover, n_e additional fictive bodies are attached to the end–effectors to allow easily description of the carried load or interaction with the environment.

The topology of the structure is defined by the vector a of $n + n_e$ integer indexes such that $a_i < i$ is the index of the previous body in the chain that is directly connected to the ith body. For instance, the tree–robot from Fig. 5 is described by

$$a = \begin{bmatrix} 0 & 1 & 2 & 2 & 4 & 3 & 5 \end{bmatrix}^T. \qquad \text{(II.3)}$$

The ith of the n_b bodies in the structure supports a Cartesian frame such that:

- the axis defined by the unit direction vector ${}^{i}z$ is the ith joint axis (the joint slides along ${}^{i}z$ if it is prismatic and turns around it if it is revolute);
- the axis defined by the unit direction vector ${}^{i}x$ is orthogonal to both ${}^{i}z$ and ${}^{a_j}z$, where $j > i$ is the index of a body directly connected to the ith body.

n_e additional frames numbered $n + 1, n + 2, \ldots, n + n_e$ are attached to the end–effectors. Their axis can be defined from the nature of the corresponding gripper or tool.

The four basic Denavit–Hartenberg parameters define the translation and orientation of each frame with respect to the previous one:

- α_i : angle between ${}^{a_i}z$ and ${}^{i}z$ around ${}^{a_i}x$;
- d_i : distance between ${}^{a_i}z$ and ${}^{i}z$ along ${}^{a_i}x$;
- θ_i : angle between ${}^{a_i}x$ and ${}^{i}z$ around ${}^{a_i}z$;
- r_i : distance between ${}^{a_i}x$ and ${}^{i}z$ along ${}^{a_i}z$.

A fifth parameter $\sigma_i \in \{0, 1\}$ is often added to specify whether the joint is revolute ($\sigma_i = 0$), in which case the joint variable q_i is equal to θ_i, or prismatic ($\sigma_i = 0$) with $q_i = r_i$.

Since the choice of ${}^{i}x$ is not unique for the bodies at the forks of a tree robot, an auxiliary axis ${}^{i,a_i}x$ is introduced, which is orthogonal to both ${}^{i}z$ and ${}^{a_i}z$. Therefore, two additional parameters are included in the *extended Denavit–Hartenberg parameters* for tree–robots:

- γ_i : angle between ${}^{a_i}x$ and ${}^{i,a_i}x$ around ${}^{a_i}z$;
- ζ_i : distance between ${}^{a_i}x$ and ${}^{i,a_i}x$ along ${}^{a_i}z$.

A graphical representation of this formalism can be found in [Domb88].

Use of Denavit–Hartenberg Parameters in Kinematics.

These parameters are introduced in $n + n_e$ 4×4 *homogeneous matrices*

$$ {}^{i,a_i}T = \begin{bmatrix} {}^{i,a_i}R & {}^{i,a_i}p \\ 0 & 1 \end{bmatrix} \tag{II.4} $$

where ${}^{i,a_i}p$ is the position of the i^{th} frame's origin and ${}^{i,a_i}R$ the 3×3 matrix of the *direction cosines* of its base vectors with respect to the previous $a_i{}^{th}$ frame [Paul81], [Domb88]. Such a matrix is obtained by

$$ {}^{i,a_i}T = \mathrm{Rot}(z, \gamma_i)\mathrm{Trans}(z, \zeta_i)\mathrm{Rot}(x, \alpha_i)\mathrm{Trans}(x, d_i)\mathrm{Rot}(z, \theta_i)\mathrm{Trans}(z, r_i), \tag{II.5} $$

where $\mathrm{Rot}(\ldots)$ denotes a homogeneous matrix corresponding to a rotation around a principal axis and $\mathrm{Trans}(\ldots)$ a homogeneous matrix of translation along a principal axis.

Homogeneous matrices allow to describe combinations of coordinates transformations performed in a sequence by a simple inner product. Considering two consecutive joints on a chain, the matrix

$$ {}^{i,a_{a_i}}T = {}^{a_i,a_{a_i}}T \cdot {}^{i,a_i}T \tag{II.6} $$

defines the *image of the i^{th} frame's base vectors in the antepenultimate frame.*

This property is extensively used to obtain the end–effectors' positions as functions of the joints for open chains mechanisms (see Section II.8).

II.6 Geometry of Manipulators with Closed Chains.

II.6.1 Parallelograms and 5–Bars Mechanisms.

Simple closed kinematic chains such as *5–bars–mechanisms or plane parallelograms* can be found in manipulators that have a dominant serial structure (see Fig. 6).

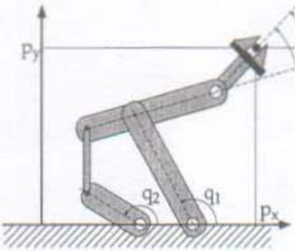


Fig.6 : 3–DOF Robot with a "5–bars Mechanism".

They allow to stiffen the manipulator and to reduce its inertia at the expense of a reduction of the workspace volume. Such manipulators can be handled as serial arms as long as their loops remain in a plane and their *free joints* are 1–DOF only [Spon89], [Domb88].

II.6.2 Parallel Robots.

In this work, *parallel manipulators* are defined as more general structures with spatial multiple or interwoven kinematic loops. They consist in *sub–chains* that connect the end–effector to the base, with at least one actuator on each sub–chain that is not shared with one. Such structures generally have many multiple–DOFs *free joints*(see Fig. 7).

The robot is said to be *fully parallel* when it consists of n_s sub–chains that have the same structure. The Delta, Hexa and Speed–R–Man robots as well as Gough's and Stewart platforms belong to this family. Others, like Pollard's structure [Poll38] are almost fully parallel since only geometrical dimensions differ between the sub–chains. Many non–fully–parallel structures have been obtained through systematic generation of articulated structures [Clav91]. However, no decisive advantage of such a mechanism over fully parallel robots could be put forward to justify the investment in developing a non–fully–parallel robot.

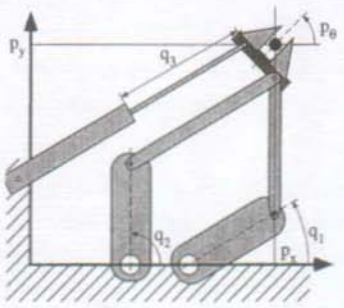


Fig.7 : Three degrees of freedom parallel robot.

II.6.3 Extension to Cooperative Robots.

Serial arms in rigid contact with their environment or with another robot (*cooperative robots*) form closed kinematic chains. They may therefore be considered as non–fully–parallel manipulators with *dynamics redundancy* since no joint is free. The redundancy makes the control of *contact forces* with the environment possible. There is therefore a strong connection between some ideas developed in the present work and results from the very active research field of cooperative robots [Dauc88a], [Dauc88b], [Chen91], [Koiv91], [Kreu89].

II.6.4 Limitations of the Denavit–Hartenberg Formalism.

The Denavit–Hartenberg modeling approach from § II.5.3 clearly corresponds to the physical structure of the manipulators it has been developed for: tree–robots and serial arms.

Extending this formalism to mechanisms with closed chains is difficult because it does not provide a systematic way to describe the closed chain's *closure constraints*. In other words, the standard modeling approach based on the Denavit–Hartenberg formalism implicitly assumes that all the joints are actuated and independent, but a satisfactory description of parallel manipulators would require a distinction between actuated and free joints.

Another drawback of the Denavit–Hartenberg approach is the large number of local coordinates systems it introduces. Some methods for dynamics modeling such as the recursive Newton–Euler algorithms were specially developed for open chains structures and alleviate this difficulty (see § III.7.2). In the general case however, the multiplication of coordinates systems makes the dynamics modeling a very tedious task.

II.6.5 Natural Coordinates.

Villalonga, Garcia de Jalon and Unda proposed to use a unique, fixed Cartesian coordinates system called *natural coordinates* to describe a mechanism's motion [Vill84], [Jalo87], [Unda87]. The position of each body is specified using the Cartesian coordinates of one or two points and one or more unitary vectors attached to it.

The points are preferably placed on joints so that different bodies can share them, and the unitary vectors are set on the joint's axis that are of interest in the model. The generalized natural coordinates vector w contains all the point's positions and the vectors's directions in Cartesian coordinates. With this representation, all the possible *constraints* can be specified under a quadratic (or linear) form.

As a very general method, this approach is applicable to parallel robots. However, it suffers from its generality since it does not provide a systematic methodology for parametrizing a given structure in a unique manner and does not exploit geometrical properties of actual manipulators. Therefore, even if dynamic modeling using *Lagrange* approach or *Kane's equations* can greatly benefit of the natural coordinates approach, thess techniques have not been used in real-time applications yet (see Section III.2).

II.7 A New Formalism for Parallel Manipulators.

II.7.1 Specificities of Parallel Manipulators.

Parallel manipulators are not general articulated mechanisms. They obey some design constraints that are imposed either by mechanical engineering, by performance optimization or by economic considerations. From the analysis of existing parallel manipulators, the following common design aspects can be identified:

- *actuated and measured joints are 1–DOF;*
- no more than two free joints can be found in each sub–chain from the manipulator's base to its end–effector;
- free joints are always 1–DOF *revolute joints*, 2–DOF *universal joints* or 3–DOF *ball–socket joints*, rarely *prismatic*.

II.7.2 A Wide Class of Parallel Manipulators.

The new model presented below for robot manipulators with closed chains can be applied to any non redundant ($n \leq 6$) structure that satisfies the two last design aspects above. The first one is a result of economic and technical considerations that were discussed in Section II.4 but does not restrict the class of manipulators that can be modeled with the proposed approach.

The second assumption is required to prevent mechanical lockings that occur in a chain of free rods. The analysis of lockings is rather involved and beyond the scope of this work but it is somewhat related to the *inverse singularities* (see § II.9.5). This restriction can therefore be understood more as a characteristic of manipulators rather than a limiting hypothesis.

The third point is the only real limitation on the class of parallel robots that can be handled with the proposed method. However, since the first goal of a robot manipulator is to position its end–effector, there is a fundamental reason to avoid free prismatic joints in a robot's structure: they prevent to transmit a driving force along a chain. Only normal forces and torques are transmitted but normal forces are unwanted because of the flexion stresses they create on the structure and the torques cause torsion stresses in the structure.

Therefore, all existing spatial parallel robots include 2–DOF *universal joints* or 3–DOF *ball–socket joints*, and plane loops in parallelograms of 5–bars mechanisms contain only free revolute joints. To the best of the author's knowledge, no existing parallel robot tincludes free prismatic joints in its structure. However, very specific devices such as the Delta's telescopic fourth degree–of–freedom combine free prismatic joints with universal joints as a torque transmission, but this mechanism can easily be modeled as an additional DOF placed in series with the manipulator and can therefore be included using the methodology proposed below.

To summarize, the considered class of structures is only a small subset of all the possible articulated mechanisms but it contains all the structures that may reasonably be used to build non–redundant parallel manipulators.

II.7.3 The Double Tree Model.

An important consequence of the design aspects from § II.7.1 is that each sub–chain that joins a manipulator's base to its end–effector has its actuated joints either:

- between the base and the first free joint;
- between the two free joints,
- between the second free joint and the end–effector.

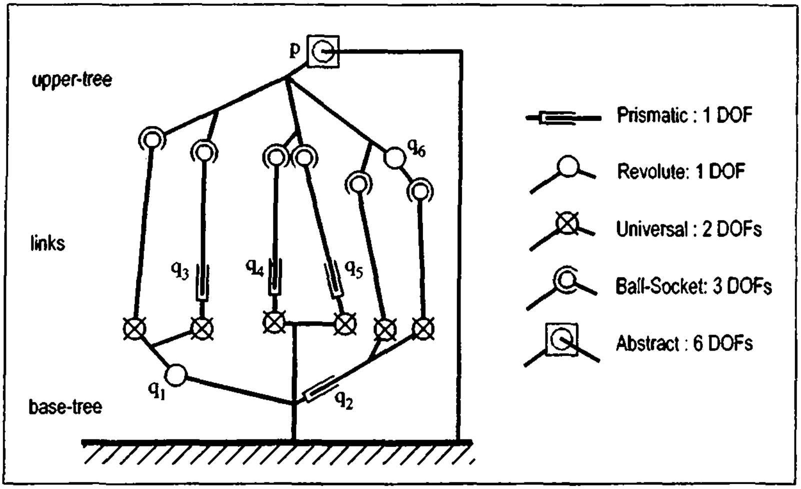


Fig.8 : A Fancy 6 DOFs Parallel Manipulator.

The first case is typical of the Delta and derived robots, the second of the Gough–platform–like structures while the third one is met when a serial structure, most often a wrist, is placed on a parallel carrier. Fig. 8 shows a fancy parallel robot with all the possible locations of the actuators. The grayed, 6–DOFs abstract joint represents the end–effector equivalent mobility. Its role is explained in § III.6.2.

II.7.4 Equivalent Structure.

The considered class of parallel manipulators can be modeled as:

- a *base–tree* attached to the base with all its ${}^{b}n$ joints actuated and 6 connection points ${}^{b}P_i$;
- *an upper–tree attached to the end–effector with all its* ${}^{u}n$ *joints actuated and 6 connection points* ${}^{u}P_i$;
- six *links* that connect both trees. The ith link connects point ${}^{b}P_i$ to point ${}^{u}P_i$. Each link has a free universal joint at one edge and a free ball–socket joint at the other edge. By convention, the universal joint is supposed attached to the base–tree. The number of actuated joints on the link is ${}^{l}n_i \in \{0, 1\}$ since two or more joints would be redundant. Note that these joints are likely to be prismatic since a revolute joint would cause unwanted flexion stresses on the links.

The whole structure therefore has amobility of

$$n = {}^{b}n + {}^{u}n + \sum_{i=1}^{6} {}^{l}n_i \,, \tag{II.7}$$

verifying the *Grübler* formula (Eq. II.2).

The *double tree* equivalent structure is based on the idea that each link removes one DOF of the upper tree relatively to the base tree by introducing the constraint

$$\left({}^{u}P_i - {}^{b}P_i\right)^2 - l_i^2 = 0, \tag{II.8}$$

where l_i is the length of the i^{th} link, which is either a constant (when ${}^{l}n_i = 0$) or a variable depending on the only allowed actuated joint on the link (${}^{l}n_i = 1$). Among all the existing parallel robots listed in Table 1, the Gough platform is the only one that uses variable–length links.

	${}^{b}n$	${}^{u}n$	${}^{l}n$
Pollard[Poll38].	3	0	0
Delta [Clav85].	3	0	0
Hexa [Pier91].	6	0	0
Speed–R–Man [Rebo91] (redundant).	6	0	0
Gough [Goug62].	0	0	1
Stewart [Stew65].	6	0	0
2 6–DOF cooperative serial arms with free spherical wrist.	1 to 6	$6 - {}^{b}n$	0
Fancy robot from Fig. 8.	2	1	$\begin{bmatrix}0 & 1 & 1 & 1 & 0 & 0\end{bmatrix}^T$

Table 1 : Dimensions of Typical Parallel Manipulators.

II.7.5 Real and Abstract Links.

Some parallel robots have an actual structure that can directly be mapped on the double tree model. The proposed formalism can also be applied to more general structures where one or more of the following situations arise:

- The link is upside–down, with a ball–socket joint at ${}^{b}P_i$ and a universal joint at ${}^{u}P_i$. As a convention, its length is then simply defined with a negative sign, which does not change the constraint relation from Eq. II.8.
- The link allows less than five DOFs of the upper tree relatively to the base tree, which is the case when it does not have a free ball–socket joint at one edge and an universal joint at the other, but more restrictive free joints. In Table 2, all the possible combinations of rotative free joints at both ends of the links are represented together with their equivalent realization with a *real link* (black) and some immaterial *abstract links* (gray). The length of the abstract links have to be defined as functions of their real link's length so that the triangle equalities are verified.

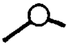
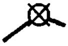
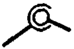
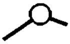
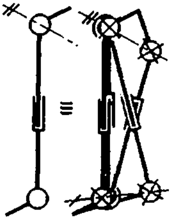
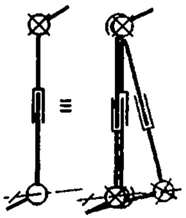
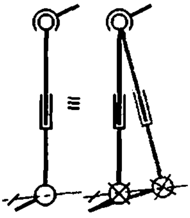
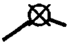
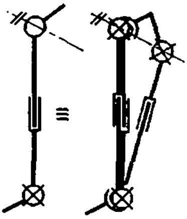
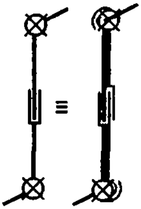
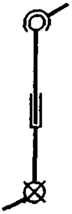
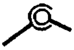
upper base	 Revolute: 1-DOF	 Universal : 2-DOFs	 Ball-Socket: 3-DOFs
	 type 4	 type 3	 type 2
	 type -3	 type 0	 type 1
	type -2	type -1	forbidden (uncontrolable DOF)

Table 2 : Table of Equivalent Abstract Links.

- The link has a null length, that is the base–tree is directly connected to the upper–tree. Three more equivalent structures correspond to these situations (see Table 3)

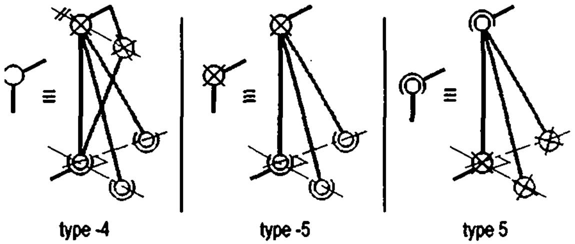


Table 3 : Null Length Equivalent Abstract Links.

II.7.6 Geometric Parametrization.

With the conventions above, the geometry of any parallel manipulator is completely specified by:

- the extended Denavit-Hartenberg parameters of its base-tree;
- the extended Denavit-Hartenberg parameters of its upper-tree;
- the type and signed length of its real links.

Since the integer numbers ${}^{b}n$, ${}^{u}n$ and ${}^{l}n_i$ are small in practice (see Table 1), the double-tree model is degenerated in most practical cases and it is therefore much easier to handle than one can expect from its formal definition. As an example, the case of the Delta robot is extensively discussed in Chapter V.

II.8 Coordinates Transformations.

II.8.1 Forward Kinematics.

The *forward kinematics* mapping defined by the function

$$\begin{aligned} &f : \mathcal{A} \rightarrow \mathcal{O} \\ &q \rightarrow p = f(q) \end{aligned} \qquad \text{(II.9)}$$

gives the position and orientation of the end-effector in the *operational coordinates system* given the manipulator's configuration, which is supposed to be uniquely obtainable from the position sensors.

Easy for Open Chains.

The composition of the transformations generated along a chain is obtained by multiplying the *homogeneous matrices* ${}^{i,a_i}T$ (see Eq. II.5):

$$\begin{aligned} {}^{n+1,0}T &= \begin{bmatrix} {}^{n+1,0}R(q) & {}^{n+1,0}p(q) \\ 0 & 1 \end{bmatrix} \\ &= {}^{1,0}T(q_1) \cdot {}^{2,1}T(q_2) \ldots {}^{n,n-1}T(q_n) \cdot {}^{n+1,n}T \quad , \\ &= \prod_{i=1}^{n+1} {}^{i,i-1}T(q_i) \end{aligned} \qquad \text{(II.10)}$$

from which f can be easily extracted.

Considering Eq. II.5, the only nonlinear functions that appear in ${}^{i,a_i}T$ are sines and cosines of q_i when the joint is revolute; ${}^{i,a_i}T$ is therefore a *smooth* (infinitely many times differentiable) function of q_i. f is therefore also smooth since it is made of products and sums of smooth functions (Eq. II.10).

Note that the modeling process above can automatically be performed using *symbolic computation* packages such as *Mathematica*™. This process is based on the recurrence Eq. II.6; therefore, the algorithm has the same structure as the manipulator itself: sequential for a serial arm or recursive for a tree–robot.

Difficult for Closed Chains.

Neither such systematic approach nor general results about the form of $f(q)$ exist for manipulators with closed chains. Geometrically, each ${}^{u}P_i$ point of the double-tree model lays on a sphere of radius l_i centered on the corresponding ${}^{b}P_i$ point. Once q is fixed, these six spheres as well as the distances between the ${}^{u}P_i$'s are known. The problem is then to find the six ${}^{u}P_i$ points.

Merlet [Merl90] showed that the solution is given by a root of a polynomial of high degree (40). This number has very recently been reduced to 24. Each root corresponds to a different *configuration* of the structure that is characterized by the same vector q. For plane 3–DOFs parallel manipulators such as the one represented in Fig. 7, the degree of the polynomial can be as high as 6. However, classes of such manipulators can be defined for which it is lower than five, thus allowing to obtain f in closed–form [LeeG86], [LeeK88a], [Goss91].

Similarly, for spatial structures the degree is reduced when the manipulator's geometry is simplified by considering superimposed free joints and/or equal lengths of some bodies. In this respect, the Delta robot may be considered as an extremely simplified spatial manipulator for which the degree is as low as two and therefore allows f to be obtained in closed form. Hervé [Herv91] showed that only one other structure exists with such a nice property.

In general, $f(q)$ must therefore be computed numerically. A *Newton–Raphson*–based algorithm that uses inverse kinematics relations and the Jacobian matrix is detailed in § II.9.6.

Another approach would be to use additional position measurements from additional sensors placed on some free joints. However, *sensory fusion* techniques are required to solve the coherency problem between the redundant measurements and to avoid *non–collocation* problems in control. In any case, it is important to note that a one-to-one mapping $\mathcal{A} \leftrightarrow \mathcal{C}$ can only be obtained for a parallel manipulator when additional information is available about its *assembly mode to select the proper solution* from the set of possible ones.

II.8.2 Inverse Kinematics.

The *inverse kinematics* mapping is defined by the function

$$\begin{aligned} & f^{-1} : O \rightarrow \mathcal{A} \\ & p \rightarrow q = f^{-1}(p) \end{aligned} \qquad \text{(II.11)}$$

Difficult for Open Chains.

Obtaining

$$q = f^{-1}(p) \qquad \text{(II.12)}$$

involves solving the nonlinear system of equations from Eq. II.9.

This system has a closed form solution only when the robot has less than 6 DOFs, or when a 6–degrees–of–freedom chain contains three consecutive prismatic joints or three revolute joints with convergent axis [Domb88]. All the current industrial robots are built to meet one of these conditions: *Cartesian robots* meet the first one and other 6–DOFs manipulators include a *spherical wrist* to satisfy the latter.

In these cases, a closed-form of f^{-1} can be obtained, which contains inverse circular functions of quotients and square roots [Domb88]. Note that these functions are not smooth, but may be considered as such in the robot's workspace, where both f and f^{-1} are one–to–one.

Simpler for Closed Chains.

For parallel manipulators, the inverse kinematics problem is brought back to the serial case since each *sub–chain* that joins the manipulator base to its end–effector may be handled as an open chain with no more than five DOFs. Moreover,

- since *free joints* in these chains are often *universal joints* or *ball–socket joints*, many Denavit–Hartenberg parameters of the sub–chains are equal to zero and computations are much simplified;
- only actuated joints are of interest, there is no need to compute anything about the free joints;
- elements of $f^{-1}(p)$ that correspond to different sub–chains are independent.

Therefore, computing the inverse kinematics of a n–DOFs parallel robot is much less complicated than for a n–DOFs serial arm. Moreover, the structure of the algorithm is intrinsically parallel, which allows to implement efficiently the algorithm on a multiprocessor control unit.

II.9 Differential Kinematics: the Jacobian and its Inverse.

II.9.1 The Jacobian Matrix of Open Chain Mechanisms.

Differentiating Eq. II.9 yields

$$dp = J(q) \cdot dq, \tag{II.13}$$

where J is the *Jacobian matrix* defined by its generic element

$$J_{i,j} = \frac{\partial p_i}{\partial q_j} = \frac{\partial f_i(q)}{\partial q_j}. \tag{II.14}$$

The Jacobian of serial manipulators can be obtained in a closed form by applying Eq. II.14 to the general formulation of $f(q)$ presented in § II.9.1. Since the form of f is known, the elements of J can only contain circular functions of q. For this reason, the Jacobian of serial manipulators is *smooth* and *bounded*. Its importance in kinematics appears when the time derivative of joint and operational positions is considered:

$$\dot{p} = J(q) \cdot \dot{q}. \tag{II.15}$$

II.9.2 Singularities in the Jacobian.

The Jacobian matrix becomes *singular* at certain manipulator configurations. In these cases, some directions of the end–effector's movements are lost since displacements caused by the joints are no more independent. This happens mainly at the workspace boundary, but also for some positions in the workspace itself, for example when the end–effector lies on a revolute joint's axis.

This situation can also be understood as a loss of mobility of the end–effector, resulting in a local redundancy of the robot. When the robot is operated close to these singularities, one may apply pseudo–inverse–based strategies developed for redundant manipulators [Colb91].

II.9.3 The Inverse Jacobian Matrix of Open Chain Mechanisms.

The *inverse Jacobian* matrix can be defined by its generic element:

$$J^{-1}_{i,j} = \frac{\partial q_i}{\partial p_j} = \frac{\partial f^{-1}{}_i(p)}{\partial p_j}. \tag{II.16}$$

This matrix is unbounded since the forward Jacobian may become singular for certain manipulator configurations.

Since all industrial serial arms have a closed form inverse kinematics relation $q = f^{-1}(p)$, their inverse Jacobian can either be obtained using Eq. II.16 or by inverting their forward Jacobian. In practice, the latter solution is often preferred and the inverse Jacobian is therefore written as a function of the joint space variables q. The kinematics relation Eq. II.15 is then written as

$$\dot{q} = J^{-1}(q) \cdot \dot{p}. \tag{II.17}$$

Note that J^{-1} is unbounded, but regular since J is bounded. Formally, J^{-1} contains non–smooth functions. However, from an intuitive point of view, it can be considered as smooth in the usable *restricted workspace* where its elements remain real and bounded.

II.9.4 The Inverse Jacobian of Parallel Manipulators.

For parallel manipulators, the inverse Jacobian matrix is derived from Eq. II.16 since the inverse kinematic relation $f^{-1}(p)$ is well defined while $f(q)$ is not. The kinematic relation between velocities in operational space and joint space takes a form which is slightly different from Eq. II.17 since the inverse Jacobian is now parametrized by operational space variables:

$$\dot{q} = J^{-1}(p) \cdot \dot{p}. \tag{II.18}$$

From a practical point of view, the independence of the sub–chains may be exploited here again to compute independent lines of J^{-1} on a multiprocessor control unit.

II.9.5 Inverse Singularities.

An amazing property of the inverse Jacobian of manipulators with closed kinematic chains is that it may become singular at some configurations. In this work, such configurations are called *inverse singularities* to distinguish them from the Jacobian' singularities described in § II.9.2.

In these configurations, some of the actuators have no more independent action on the manipulator or, in other words, the elements of $\dot{q}$ are no more independent. The robot is then not controllable any more since the actuators can no more move the end–effector along certain directions.

However, contrarily to the case of mobile robots described in § II.3.5, the friction constraints in a manipulator's structure are generally not strong enough to prevent an actual displacement of the end–effector to occur along the forbidden directions. In those configurations, the robot becomes uncontrollable since some displacements of the end effector are possible without any action on the actuators. The robot's state may even become unobservable when only actuated joint variables are measured.

The inverse Jacobian may become singular inside of the workspace. This singularity is known to occur when at least three links of the double tree model become parallel or coplanar. A very challenging *problem is to find all these configurations for a given manipulator structure.* Searching for the roots of the inverse Jacobian's determinant is a very tedious task, even with intensive use *symbolic computation* packages such as *Mathematica*™. However, some results were obtained for Gough's platform structures using advanced mathematical tools such as Plücker's lines and Grassmann geometry [Merl89].

An even more important question is: "How to design a parallel robot such that its inverse singularities are outside a given workspace?" [Clavel, 1993, personal communication]. While some numerical results about the geometry of Gough's platforms were obtained [MaO91], the general problem is much more difficult when the structure itself has to be determined with its assembly mode and associated joint space.

II.9.6 Jacobian of Parallel Robots.

Since f has no closed form, it is definitely impossible to obtain a closed form of the Jacobian of a general parallel manipulator as a function of the joint position q. The only way to analytically obtain it is to invert the manipulator's inverse Jacobian and therefore obtain it as a function of the operational position p:

$$J(p) = [J^{-1}(p)]^{-1}. \tag{II.19}$$

Contrarily to serial arms, the Jacobian matrix of a parallel manipulator is unbounded for some values, since the inverse Jacobian may become singular; this has very important consequences in dynamics (see Chapter III).

For practical applications, no other choice is left than numerically inverting $J^{-1}(p)$ if the Jacobian matrix is required in a parallel robot's control unit because the closed form is too complicated to allow an efficient implementation.

Application to Forward Position Kinematics.

The basic idea is to start from the measured q, an initial approximation of p denoted ${}^{0}p$ and the Jacobian matrix ${}^{0}J$ and to compute

$$\begin{cases} \Delta q = f^{-1}({}^{k-1}p) - q \\ {}^{k}p = {}^{k-1}p + {}^{k-1}J \cdot \Delta q \\ {}^{k}J = [J^{-1}({}^{k}p)]^{-1} \end{cases} \tag{II.20}$$

for $k = 1, 2, \ldots$ until a desired precision is reached, for example when

$$\Delta q < \varepsilon, \tag{II.21}$$

where ε is given by the resolution of the joint position sensors.

The re–evaluation of the Jacobian in the last line of Eq. II.20 is in fact necessary only when elements of ${}^{0}J$ are so close to zero that sign changes may be suspected. In practice, the p and J obtained during the previous sampling interval can be used as inital values. Since the robot is supposed to move only a little during a sampling interval, three or four iterations are usually enough to reach the precision criterion defined in Eq. II.21.

II.9.7 Extended Jacobian Matrix and Absolute Accuracy.

A common problem for the designer of a manipulator is to obtain the maximal error on the end–effector's position as a function of the tolerances on all the mechanical parts. This is known as the *absolute accuracy* problem.

When all the nominal dimensions are included in an extended joint position vector $\bar{q}$ that also contains the articulated joint variables q, one can obtain an extended forward kinematic relation

$$p = \bar{f}(\bar{q}), \; p \in \mathcal{O}. \tag{II.22}$$

An *extended Jacobian* can then be obtained as

$$J_{i,j} = \frac{\partial p_i}{\partial \bar{q}_j} = \frac{\partial \bar{f}_i(\bar{q})}{\partial \bar{q}_j}. \tag{II.23}$$

The maximal error of the end–effector's position along the i^{th} dimension of the operational space is then given by

$$\max(\delta \bar{p}_i) = \max(J_i(\bar{q}) \cdot \delta \bar{q}). \tag{II.24}$$

The problem is not trivial since the maximization should be performed over the complete joint space and for all the possible $\delta\bar{q}$ vectors that contain combinations of the negative and positive maximal tolerances of each dimension. For serial robots however, the problem is made easier since the boundedness and smoothness properties of the Jacobian matrix can be exploited.

The absolute accuracy problem is much more involved for parallel manipulators, for which none of these advantages holds. First, the extended inverse Jacobian only has a closed form when each sub-chain is parametrized by at most six elements of $\bar{q}$:

$$J^{-1}_{i,j} = \frac{\partial \bar{q}_i}{\partial p_j} = \frac{\partial f_i^{-1}(p)}{\partial p_j}. \tag{II.25}$$

Second, if J^{-1} is rectangular, the extended Jacobian can only be obtained through a pseudo-inversion, which lacks a physical significance.

Therefore, it is easier to address the problem from a reversed point of view: find the mechanical tolerances that guarantee a given maximal end-effector position error using the relation

$$\max(\delta\bar{q}_i) = \max\left(J^{-1}(\bar{q}) \cdot \delta\bar{p}\right). \tag{II.26}$$

A very annoying consequence appears : at the *inverse singularities*, the slightest mechanical error is amplified and causes very large errors of the end-effector's position. Hence, the solution of the absolute precision problem for parallel robots depends on the definition of a *restricted workspace* in which the maximization has to be performed.

II.9.8 Acceleration Kinematics.

For Open Chain Structure.

Differentiating Eq. II.14 with respect to time gives the generic element of the time derivative of the Jacobian matrix

$$\dot{J}_{i,j}(q,\dot{q}) = \frac{d}{dt}\frac{\partial f_i(q)}{\partial q_j} = \frac{\partial}{\partial q_j}\left(\frac{d}{dt}f_i(q)\right) = \frac{\partial}{\partial q_j}\left(\sum_{k=1}^{n}\frac{\partial f_i(q)}{\partial q_k}\cdot\dot{q}_k\right). \tag{II.27}$$

Similarly,

$$\dot{J}^{-1}(p,\dot{p}) = \frac{\partial}{\partial p_j}\left(\sum_{k=1}^{n}\frac{\partial f_i^{-1}(p)}{\partial p_k}\cdot\dot{p}_k\right). \tag{II.28}$$

Differentiating Eq. II.17 gives the acceleration of the end–effector

$$\ddot{p} = J(q) \cdot \ddot{q} + \dot{J}(q, \dot{q}) \cdot \dot{q} . \tag{II.29}$$

An inverse relation can be obtained from Eq. II.29 at the expense of the manipulator's Jacobian inversion, which is in practice computed numerically:

$$\ddot{q} = J^{-1}(q) \cdot [\ddot{p} - \dot{J}(q, \dot{q}) \cdot \dot{q}] . \tag{II.30}$$

Structures with Closed Chain.

Differentiating Eq. II.18 with respect to time gives the accelerations of the joints

$$\ddot{q} = J^{-1}(p) \cdot \ddot{p} + \dot{J}^{-1}(p, \dot{p}) \cdot \dot{p} . \tag{II.31}$$

Through numerical inversion of the inverse Jacobian matrix, a direct relation can be obtained:

$$\ddot{p} = J(p) \left[\ddot{q} - \dot{J}^{-1}(p, \dot{p}) \cdot \dot{p}\right] . \tag{II.32}$$

II.10 Desired Task and Generation of Reference Signals.

II.10.1 Task Description.

In Chapter I, the task was defined from a very general point of view, without any reference to time. In this work, the considered tasks are paths defined in the n-dimensional operational space that the robot's end-effector should track as closely as possible. The user is supposed to define the path's geometry as a function $p_c(s)$ of the curvilinear abscissa s and to specify its length s_f.

II.10.2 The Profile Function.

The reference signals generation requires to introduce a *profile function* $s(t)$ for $0 \le t \le t_f$ such that $s(0) = 0$ and $s(t_f) = s_f$ that gives the curvilinear abscissa as a function of time:

$$p_c(t) = p_c(s)\Big|_{s(t)}, \text{ for } 0 \le t \le t_f,$$

$$\dot{p}_c(t) = \frac{\partial}{\partial s}p_c(s)\Big|_{s(t)} \cdot \dot{s}(t),$$

$$\ddot{p}_c(t) = \frac{\partial^2}{\partial s}p_c(s)\Big|_{s(t)} \cdot \dot{s}^2(t) + \frac{\partial}{\partial s}p_c(s)\Big|_{s(t)} \cdot \ddot{s}(t). \tag{II.33}$$

In most robotics applications, the profile function is not of interest for the user, who only requires some constraints on velocities and accelerations to be respected. Ideally, the profile function should be automatically determined in order to minimize the task's duration under the specified constraints [Bobr88].

However, none of the existing results allows to perform this optimization in real-time, which limits their use to the case of very repetitive tasks. An attractive idea is to consider the robot as constrained on the trajectory, in which case its kinematics and dynamics can be written as functions of the curvilinear absciss only [Dahl92]. In this case, the complete robot control problem becomes SISO and on-line travel-time minimization techniques could be developed [Dahl90]. In this work, since the paths to be tracked are not supposed to be known in advance, fixed profile functions are used.

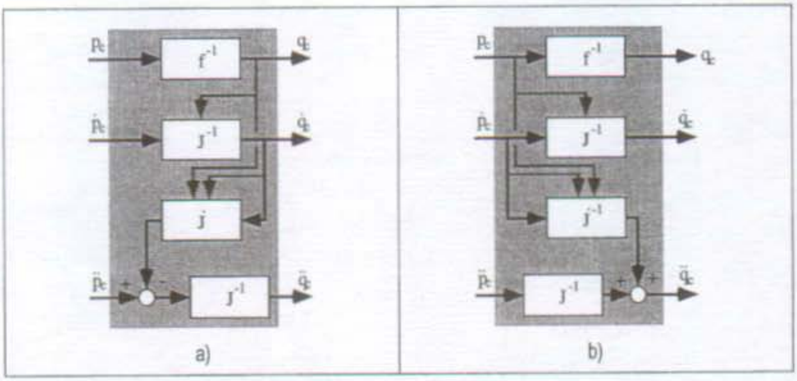
II.10.3 Conversion to Joint Space.

When a joint space control strategy is chosen (see § I.6.1), conversion of the reference signals into joint space is required and the inverse kinematics relations are used for this purpose.

However, it is a common habit in practice to implement only Eq. II.11 to convert the reference positions and then to use numeric differentiation routines to obtain velocities and accelerations. In this case, it is no more useful to compute Eq. II.17 and Eq. II.30. This shortcut in the reference generation result in a damaging of the signals' quality since numerical errors in the implementation of the position inverse kinematic relation Eq. II.11 are amplified by the numerical differentiation. Such errors are non negligible since inverse circular functions are involved in the f^{-1} function. They may however be reduced by using a high–precision representation of the real numbers in the control unit [Codourey, personal communication, 1991]. Nevertheless, this shortcut should be avoided in high–performance robot control since the reference acceleration signal plays a crucial role in the feedforward control generation (see Chapter IV).

Different Schemes for Serial and Parallel Manipulators.

For a serial arm, the complete reference conversion algorithm should compute Eq. II.11, Eq. II.17 and Eq. II.30 according to the scheme in Fig. 9a. Note that the parameters of each block are indicated by



a) b)

Fig.9 : Conversion of Reference Signals.

grayed arrows. The corresponding flow of data is sequential: q_c must be available in order to compute J^{-1} and $\dot{q}_c$ is required before $\dot{J}$ can be evaluated.

For a parallel manipulator, the scheme is different since Eq. II.11, Eq. II.18 and Eq. II.31 are used. The resulting scheme is given in Fig. 9b. In this case, the flow of data is intrinsically parallel: almost all the computations can be performed simultaneously.

II.11 Conclusions.

The differences between parallel and serial manipulators and their kinematics models are summarized in Table 4 below.

	Serial arms.	Parallel manipulators.
All joints measured.	Yes.	No.
All joints actuated.	Yes.	No.
Systematic description of the structure.	Yes, Denavit–Hartenberg.	New! 'Double–Tree' model fromSection II.7.
Forward kinematics.	Yes, easy.	No general closed–form solution. Numerical algorithms in practice.
Inverse kinematics.	No general solution, yes in practical cases.	Yes, easy in practice.
Jacobian matrix J	Function of q, easy to obtain in closed form. Bounded.	Function of p, no usable closed form, obtained by inversion of J^{-1}.
Jacobian singularities. Robot becomes redundant.	Yes.	Yes.
Inverse Jacobian J^{-1}	Function of q. Complicated closed form.	Function of p, easy to obtain in closed form.
J^{-1} singularities. Robot looses its stiffness.	No.	Yes.
Structure of the reference signals conversion algorithm.	Sequential.	Parallel.

Table 4 : Comparative Kinematics: Serial versus Parallel Structures.

The advantages of serial arms rely on their very simple structure where each joint is both measured *and actuated. Their kinematics is then easily described using forward relations, where the corresponding* algorithms have an intrinsically sequential structure. Parallel manipulators require a different modeling approach which is based on inverse kinematics relations, with algorithms that exhibit a fundamentally parallel nature.

Chapter III

Manipulator Dynamics.

III.1 Introduction.

The dynamics of ideally rigid and frictionless manipulators are considered in this Chapter. As in the previous one, special attention is paid to the differences between serial and parallel mechanisms. The various approaches to the problem of modeling the dynamics of a system of rigid bodies are first introduced in Section III.2. Their field of application in robotics is also mentioned.

Energy based methods are introduced in Section III.3 with the principle of virtual work. Their application to manipulators shows the key importance of the Jacobian matrix in dynamics. As a first consequence of this fact, it is shown that parallel manipulators may not be as stiff as they are often regarded.

The Lagrange approach to dynamics is presented in Section III.4 as a general method that explicitly considers constraints and allows dynamics' analysis. The specific case of open chain mechanisms is dealt with in Section III.5 to introduce the concepts of inverse dynamics models in both joint space and operational space.

In Section III.6, dynamics of parallel manipulators are discussed. It is shown that the elimination of Lagrange multipliers is equivalent to solving a linear system of equations where the system's matrix is the manipulator's inverse Jacobian. Next, a new formulation of the inverse dynamic model of a parallel robot is developed, based on the double tree model. The resulting dynamics model is called "in the two spaces".

Algorithms to compute efficiently inverse dynamics models of manipulators are considered in Section III.7. First, the well–known Newton–Euler algorithm for tree robots is reminded and extended to perform some useful kinematics computations. Then, a new algorithm is developed to evaluate systematically the inverse dynamics model of any parallel robot described by the double–tree kinematics model. Its complexity is shown to be of the same order of magnitude as the standard algorithm for open–chain manipulators.

In Section III.8, a general formulation of the dynamics of all types of manipulators is derived from the model in the two spaces. It is an inverse dynamics model in operational space only. Some important properties of the dynamics of parallel robots are then established. The most important result is that passive mappings can be determined only locally. Then a comparison is made between dynamics of serial arms and parallel robots. The common belief that parallel robots have lower inertia than serial arms is shown to be unjustified if the restricted workspace of the robot is not correctly defined.

Finally, the original contributions discussed in this chapter are summed up in Section III.9.

III.2 Rigid Body Dynamics: an Overview.

Over the two past centuries, many methods of formulating the dynamics equations of systems of rigid bodies have been developed. They can be classified as either *Newton–Euler* or energy methods. The latter include *Lagrange, Hamilton,* and *Kane's equations* [Kane85] as well as more recent results based on *screw theory* [Penn92],[Pete92] or *motor algebra* [Sugi87],[Sugi89]. They all eliminate the need to identify the constraint forces and therefore somewhat rely on the very fundamental *principle of virtual work* detailed in Section III.3.

III.2.1 Newton–Euler for Control.

In the Newton–Euler approach, all forces applied to a rigid body are identified and set equal to the inertial forces. Its main drawback is the description of constraint forces which, especially in three–dimensional problems, becomes a tedious task.

For model-based control, an *inverse dynamics model* of the robot should be obtained to compute the actuator forces and torques from the desired acceleration, given the robot's *state* that contains positions and velocities. The Newton–Euler method provides such models that can be computed very efficiently for open–chain structures. This standard algorithm is reviewed in § III.7.2 since it is used in the new algorithm for the computation of the inverse dynamics model of parallel robots proposed in § III.7.4.

III.2.2 Lagrange for Analysis and Control.

A Variational Approach.

Lagrangian mechanics is certainly the most popular energy-based approach. Kinetic and potential energies are specified and variational calculus is used to determine the dynamic equations of motion. The constraints are eliminated through minimization of a definite integral using *Lagrange multipliers*. This has often been considered as a major drawback of the approach since Lagrange multipliers have no physical meaning in the general case. However, it is shown in Section III.6 that a very interesting physical meaning can nevertheless be found and used when parallel robots are considered.

A Tool for Analysis.

It is undisputed that Lagrange and Hamilton approaches are fundamental tools for dynamic analysis of mechanical systems. Once the rather abstract modeling process is finished, the resulting inverse dynamic model may be put in a very compact symbolic form. Moreover, the model is physically meaningful since geometrical dimensions and inertial parameters explicitly appear in the formulation together with positions, velocities and acceleration (or momenta) variables.

An Efficient Approach to Model-Based Control.

On the contrary, the use of the inverse dynamics model obtained through this method for the model-based control has been much debated. For serial arms, it has been recently been proved to be as efficient as the algorithms based on the Newton–Euler approach (see § III.7.2). For parallel manipulators, no quantitative comparison has yet been published, but the elimination of Lagrange multipliers was commonly considered as a poorly efficient method. The result from Section III.6 below shows that this opinion is unjustified when one considers that kinematics relations have any way to be included in the controller.

III.2.3 Modern Theory and Simulation.

Modern approaches such as *Kane's equations* [Kane85] are based on the principle of virtual works; they have been developed to alleviate problems met with the variational approach. These methods provide a way to visualize externally applied forces, but they also require the differentiation of constraint functions, which are not given any geometric significance. More recent techniques based on the *screw theory* [Penn92],[Pete92] or on the *motor algebra* [Sugi87],[Sugi89] rather multiply the constraints by an *orthogonal complement* [Ange88] to eliminate them. They are indeed appropriate to model large, complex mechanisms. However, they do not exploit the specificities that distinguish an actual parallel manipulator from a general articulated structure and therefore lack the analytical advantages of the variational approaches. Moreover, Kane's approach requires a numerical integration of the constraint equations in the general case, explaining probably why no model based-control has been implemented using this method. However, early results such as Ait-Ahmed's thesis [Ait93] contribute to reduce these drawbacks.

Simulation Techniques.

For *simulation* purposes, a *forward dynamic model* of the manipulator should be available to compute the manipulator's positions, velocities and accelerations corresponding to given actuator torques and forces. Such models can be derived from inverse models using the *composite rigid body method* of

Walker and Orin [Walk82]. Featherstone [Feat87] proposed the *articulated body method* as a better conditioned, more efficient alternative. Again, these methods implicitly use the serial structure of traditional robot arms and may not be applied to parallel manipulators without extra care.

Today, the most widely used approach in simulation is based on *Kane's equations* expressed in *natural coordinates* (see § II.6.5). Simulation software packages such as *Mechanica*™ use symbolic computations to build models with this technique and numerically integrating them using sophisticated algorithms.

III.3 Virtual Work Principle.

A very fundamental result in mechanics is the *principle of virtual work* of *d'Alembert*. One of its consequence is that the virtual works of any set of generalized forces system that causes the same *virtual displacement* of the structure are equal. Virtual displacements differ from actual displacements in that they must only satisfy the structure's geometry and no other law of motion: the principle does not refer to any geometric or inertial parameter and may therefore be considered as the fundamental connection between dynamics and kinematics.

Since any generalized forces system can be used, the equality of the virtual works associated to the the two coordinates systems of interest in robotics yields:

$$\tau^T \cdot \delta q = \upsilon^T \cdot \delta p, \qquad \text{(III.1)}$$

where τ is the force/torque vector that corresponds to joint space virtual displacement δq and υ can be thought of as forces/torques applied on the robot's end-effector that cause the δp virtual displacement in operational space.

When the differential kinematics relation from Eq. II.13 is introduced in Eq. III.1, the importance of the Jacobian matrix in dynamics clearly appears in a very fundamental relation in robot dynamics since it holds for any structure:

$$\tau = J^T \cdot \upsilon \qquad \text{(III.2)}$$

III.3.1 Application to Static Stiffness Determination.

As a first application of Eq. III.2, the stiffness of a manipulator is discussed. The actuators are supposed rigidly fixed, maintaining the robot in a constant configuration. A force υ applied on the end–effector causes a small deplacement δp of it such that

$$\upsilon = K \cdot \delta p, \tag{III.3}$$

where K is the manipulator's *stiffness matrix* as seen from the end-effector, which depends on the configuration.

It is assumed here that all the flexibility in the structure may be modeled as linear springs at the actuated joints by

$$\tau_i = k_{e,i} \delta q_i, \; i = 1 \ldots n. \tag{III.4}$$

This assumption is realistic as long as the deformations of the bodies in the structure are small, which allows a first order approximation to be valid.

Using Eq. II.16, Eq. III.2, Eq. III.3 and Eq. III.4, the stiffness matrix is given by

$$K = J^{-T} \cdot \begin{bmatrix} k_{e,1} & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & k_{e,n} \end{bmatrix} \cdot J^{-1}. \tag{III.5}$$

Parallel robots are often considered stiffer than serial robots because their $k_{e,i}$ constants are much higher because some bodies in their structure are stressed in traction and compression rather than flexion. However, Eq. III.5 shows that parallel robots have a structural fundamental drawback since K becomes singular at the *inverse singularities*, indicating that the robot's stiffness is completely lost (see § II.9.5).

This is discussed in § V.4.2 where the Delta's workspace is defined as the subspace of *O* that is "far enough" from the inverse singularities to guarantee a minimal stiffness.

III.4 Lagrange Method.

III.4.1 A Very General Approach.

The most used approach of modeling articulated mechanical structures is based on the Lagrangian $e_L = e_K - e_P$ where e_K is the system's *kinetic energy*, and $e_P(w)$ is its *potential energy*. w is any m-dimensional vector of generalized coordinates that allow to specify the structure's dynamics *state* in an unique manner.

The i^{th} generalized force that should be applied on the structure is given by

$$\varsigma_i = \frac{d}{dt}\left(\frac{\partial e_L}{\partial w_i}\right) - (B_i^T \cdot \lambda) \text{ for } i = 1 \ldots n. \qquad \text{(III.6)}$$

Since the structure has only a mobility $n \le m$, n components of the generalized coordinates are independent. Therefore only n generalized forces from ς can be exerted on the structure, the others being set to zero. The nonzero elements of ς are grouped in the τ vector of forces and torques that are applied on the manipulator through the n actuators.

Furthermore, $m - n$ independent *constraints* between the m generalized coordinates can be written under the form:

$$b_i(w) = 0, i = 1 \ldots m - n, \qquad \text{(III.7)}$$

and the $(m - n) \times m$ matrix B of the partial derivatives of the constraints obtained as

$$B_{i,j} = \frac{\partial b_i}{\partial w_j}, \text{ for } i = 1 \ldots m - n, j = 1 \ldots m. \qquad \text{(III.8)}$$

Hence, Eq. III.6 can be viewed as a linear system of m equations with m unknown variables $\{\tau, \lambda\}$, the vector λ grouping the $m - n$ *Lagrange multipliers*.

General Form of the Dynamics Equation.

The kinetic energy is a quadratic function of the generalized velocities of the form

$$e_K(w, \dot{w}) = \frac{1}{2}(\dot{w}^T \cdot A(w) \cdot \dot{w}), \qquad \text{(III.9)}$$

where $A(w)$ is a $m \times m$ *symmetric* and *positive definite* matrix.

Since the potential energy is independent of the velocities, Eq. III.6 is a nonlinear differential algebraic equation of the second order

$$\varsigma = A(w) \cdot \ddot{w} + \Gamma(w) \cdot \dot{w} \cdot \dot{w} + g(w) - (B^T \cdot \lambda) \qquad \text{(III.10)}$$

where $g(\dot{w})$ is the vector with $g_i = \frac{\partial e_P}{\partial w_i}$ and Γ is a tensor containing the *Christoffel symbols*

$$\Gamma_{i,j,k} = \frac{1}{2}\left(\frac{\partial A_{k,j}}{\partial w_i} + \frac{\partial A_{k,i}}{\partial w_j} - \frac{\partial A_{i,j}}{\partial w_k}\right). \qquad \text{(III.11)}$$

To avoid introducing tensor calculus, Eq. III.10 is usually written under the purely matrix form

$$\varsigma = A(w) \cdot \ddot{w} + C(w, \dot{w}) \cdot \dot{w} + g(w) - (B^T \cdot \lambda) \qquad \text{(III.12)}$$

where C is a $m \times m$ matrix defined by its general element

$$C_{i,j} = \sum_{k=1}^{m} \Gamma_{i,j,k} \cdot \dot{w}_k. \qquad \text{(III.13)}$$

III.4.2 Physical Parameters.

The general form of the dynamics from Eq. III.10 has the important property of being *linear in the parameters* [Spon89], i.e. it can be written as

$$\tau = Y(w, \dot{w}, \ddot{w}) \cdot \varphi, \qquad \text{(III.14)}$$

where φ is a vector of n_φ expressions containing only geometric and inertial parameters of the manipulator and Y is a $n \times n_\varphi$ matrix of functions of w, $\dot{w}$, $\ddot{w}$ only.

It should be stressed that this parametrization is not unique: it is possible to find many $\{\varphi, Y\}$ pairs that satisfy Eq. III.14. Khosla [Khos89a] showed that the number of independent parameters of φ is always larger than the number of physical parameters of the manipulator . It is therefore impossible to identify each of them independently, and this has important consequences in adaptive control (see § IV.3.6).

Geometrical parameters such as lengths and radii are known with an extreme precision from the design stage and masses can be measured before assembling the manipulator. Only inertia momentums are possibly unknown and may therefore require an off-line identification procedure. All these parameters can then be considered as constant during the manipulator's lifetime. However, when the robot carries varying loads as in *pick & place* tasks, the quickly varying mass of the carried load affects most of the elements of φ.

III.5 Open Chain Manipulators.

III.5.1 Inverse Dynamics Model.

Since the kinetic energy and the potential energy of serial arms and tree–robots can be written as functions of the $q, \dot{q}$ joint variables only, the generalized coordinates can be chosen as $w = q$. m is then equal to n and since no constraints on the generalized coordinates appear, the dynamics equation of serial arms reduces to

$$\tau = {}^qA(q) \cdot \ddot{q} + {}^qC(q, \dot{q}) \cdot \dot{q} + {}^qg(q) . \tag{III.15}$$

Eq. III.15 is an inverse dynamics model since it gives the input torques/forces τ to apply to the structure as functions of the output acceleration $\ddot{q}$. Through inversion of ${}^qA(q)$, Eq. III.15 can be transformed in a direct model under the state–space representation that is customary in control theory. In this context, q and $\dot{q}$ clearly appear as the manipulator's *state variables*. The state–space representation is not used in robotics since analysis and model–based control are much easier with the inverse model from Eq. III.15.

The Inertia Matrix.

${}^qA(q)$ is known as the manipulator's *inertia matrix*. It contains inertias on its diagonal and the non–diagonal terms hold *cross–inertias* that reflect the *couplings* between the structure's degrees–of–freedom.

Quadratic Forces.

The expression ${}^qC(q, \dot{q}) \cdot \dot{q}$ describes the so–called *quadratic forces* since products of generalized velocities are involved. Terms that contain a $\dot{q}_i^2$ correspond to *centripetal forces* and those with a $\dot{q}_i\dot{q}_j$ for $i \neq j$ denote *Coriolis* effects. ${}^qC(q, \dot{q})$ is often called the *quadratic forces matrix*.

Gravity Forces.

Finally, the ${}^qg(q)$ vector usually describes the effects of the *gravity* on the structure. Note that this term can be included in the previous ones by considering the base of the robot as uniformly accelerated. This is implicitly the case in the *Newton–Euler* approach.

A Very Important Property: Passivity.

From Eq. III.11 and Eq. III.13, one can observe that ${}^q\dot{A}(q) - 2{}^qC(q, \dot{q})$ is *skew–symmetric* [Orte89]. This leads to a very important property of the dynamics of serial manipulators known as *passivity*, which has extensively been used in the design of robust controllers as well as in the proofs of convergence of adaptive schemes (see § IV.5.3). This property is shown to hold also for parallel manipulators in § III.8.2 and its physical meaning is discussed in § III.8.3.

III.5.2 Formulation of the Model in Operational Space.

As defined in § I.3.1, a robot manipulator is supposed to perform a task with its end–effector. One may therefore feel disconcerted by the model from Eq. III.15 where no reference to operational space is made. Since the task and performance criteria are specified in operational space and the input torques/forces are applied at the joints, inverse kinematics clearly must appear somewhere in a dynamics model that describes the input–output behavior from the user point of view. Introducing the inverse kinematics relations Eq. II.15 and Eq. II.30 in the inverse dynamics model in joint space Eq. III.15 yields

$$\tau = {}^{q}A(q) \cdot [J^{-1}(q) \cdot [\ddot{p} - \dot{J}(q,\dot{q}) \cdot J^{-1}(q)\dot{p}]] + {}^{q}C(q,\dot{q}) \cdot J^{-1}(q)\dot{p} + {}^{q}g(q) . \qquad \text{(III.16)}$$

The various matrices in Eq. III.16 are functions of the manipulator's state $\{q, \dot{q}\}$. Eq. II.12 and Eq. II.17 can then be used to substitute operational state variables $\{p, \dot{p}\}$ for the parameters if Eq. II.11 is one–to–one in the considered workspace. This parameter substitution is indicated by a prime symbol in the following developments. Note that any matrix with a prime has the same numerical value as the original matrix for a given manipulator state given by a {position, velocity} pair, but their symbolic representation differs. The resulting inverse dynamics model is then

$$\tau = M(p) \cdot \ddot{p} + D(p,\dot{p}) \cdot \dot{p} + h(p) , \qquad \text{(III.17)}$$

with

$$M(p) = {}^{q}A'(p) \cdot J'^{-1}(p) = {}^{q}A(f^{-1}(q)) \cdot J^{-1}(f^{-1}(q)), \qquad \text{(III.18)}$$

$$D(p,\dot{p}) = {}^{q}A'(p) \cdot \dot{J}'^{-1}(p,\dot{p}) + {}^{q}C'(p,\dot{p}) \cdot J'^{-1}(p) \qquad \text{(III.19)}$$

$$h(p) = {}^{q}g'(p) . \qquad \text{(III.20)}$$

Note that the manipulator's state is now defined in operational space as $\{p, \dot{p}\}$.

The $M(p)$ is a *pseudo–inertia matrix* since the (i, j) th element contains the projection of the manipulator's inertia along the ith dimension of operational space on the jth actuator. For similar reasons, the $D(p, \dot{p})$ matrix is a *pseudo–quadratic forces matrix*.

It is important to realize that the formulations in joint space (see Eq. III.15) and in operational space (see Eq. III.17) are absolutely equivalent from the dynamics point of view as long as the kinematic mapping f is one–to–one, or in other words, as long as the Jacobian remains regular. The difference between both formulations is mainly that the joint space formulation describes the manipulator's motion using easily measurable state variables while the operational space model is more related to the end–effector's motion in the user–defined reference coordinates system.

The latter formulation of the dynamics was first proposed by Khatib in 1980 [Khat80] to model also *contact forces* between the end-effector and the environment. An important contribution of this work is to show that this formulation is also well suited to analyze and control parallel manipulators.

III.6 Parallel Manipulators.

III.6.1 Relation between the Lagrange Multipliers and the Jacobian Matrix.

In early attempts to model manipulators with closed chains, the structure is described as a single tree with closure constraints. Two distinct approaches to solve the constraints have been proposed:

- Nakamura and Ghodoussi introduced the Jacobian of the free joints with respect to the actuated ones [Naka89]. The motion of the free joints is therefore explicitly considered, which makes this approach well suited for simulation (see § III.2.3). Kokkinis and Stoughton [Kokk91b] showed its equivalence with the orthogonal complement formulation from Angeles and Lee [Ange88].
- Luh and Zheng [Luh85] and Kleinfinger and Khalil [Klei86] simultaneously proposed identical solutions based on Lagrange multipliers. This approach is now very popular [Feat87], [Domb88], [Kreu88], [Mill92]. However, these authors did not realize the equivalence of the Lagrange multipliers' elimination with the computation of the manipulator's Jacobian matrix that is established below.

Other approaches have been proposed for special cases such as Gough's platforms by Do [Do88], Geng [Geng92], Helinski [Heli90], Ji [Ji93], Liu et al. [Liu91], [Liu93] and also on parallel wrists by Cox [Cox89] or Fujimoto[Fuji91].

Generalized Coordinates.

Since the relation $p = f(q)$ is not known in closed-form (if it exists...) for a general parallel manipulator, the kinetic and potential energy of the body that carries the end-effector can not be written as closed-form functions of the joint space positions q and velocities $\dot{q}$ (see § II.8.1). This is the case for all the bodies composing the robot's upper-tree and the links.

Therefore, it is proposed here that choosing $w = \{q_1 \dots q_n, p_1 \dots p_n\}$ as the vector of generalized coordinates makes it possible to develop an inverse dynamics model of any parallel manipulator *in a very systematic manner because the n constraints can be written under the very* standard form

$$b_j(w) = q_j - f_j^{-1}(p) = 0 \tag{III.21}$$

since the inverse kinematics relations Eq. II.11 are known and available in closed form. Therefore, the inverse Jacobian matrix explicitely appears in tthe B matrix of the constraints that results from Eq. III.8:

$$B = \begin{bmatrix} I & | -J^{-1} \end{bmatrix}. \tag{III.22}$$

Elimination of the Lagrange Multipliers.

Eq. III.12 can be split in two parts

$$\begin{cases} \tau = {}^{q}A(w) \cdot \ddot{w} + {}^{q}C(w, \dot{w}) \cdot \dot{w} + {}^{q}g(w) - \lambda \\ -J^{-T} \cdot \lambda = {}^{p}A(w) \cdot \ddot{w} + {}^{p}C(w, \dot{w}) \cdot \dot{w} + {}^{p}g(w) \end{cases} \tag{III.23}$$

where

$$A(w) = \begin{bmatrix} {}^{q}A(w) \\ {}^{p}A(w) \end{bmatrix}, C(w, \dot{w}) = \begin{bmatrix} {}^{q}C(w, \dot{w}) \\ {}^{p}C(w, \dot{w}) \end{bmatrix} \text{ and } g(w) = \begin{bmatrix} {}^{q}g(w) \\ {}^{p}g(w) \end{bmatrix}. \tag{III.24}$$

For given w, $\dot{w}$ and $\ddot{w}$, the terms in the right-hand side of Eq. III.23 are constant vectors and J^{-T} is a constant matrix. Therefore, the Lagrange multipliers λ can be found by solving the linear system corresponding to the second line of Eq. III.23, or symbolically

$$\lambda = -J^{T}(p)\left[{}^{p}A(w) \cdot \ddot{w} + {}^{p}C(w, \dot{w}) \cdot \dot{w} + {}^{p}g(w)\right] \tag{III.25}$$

that is defined for every p *such that J^{-1} is regular (see § II.9.5).*

III.6.2 Inverse Dynamic Model in the Two Spaces.

Assume that the kinetic energy and potential energy of each of the structure's bodies can be expressed either using operational space coordinates p or using joint space coordinates q. This is not a restrictive hypothesis since any undesirable reference to q or $\dot{q}$ can be removed by using the inverse kinematics relations Eq. II.11 and Eq. II.17.

Hence, the inertia matrix, the quadratic forces matrix and the gravity vector are respectively:

$$A(w) = \begin{bmatrix} {}^{q}A(q) & 0 \\ 0 & {}^{p}A(p) \end{bmatrix}.$$

$$C(w, \dot{w}) = \begin{bmatrix} {}^{q}C(q, \dot{q}) & 0 \\ 0 & {}^{p}C(p, \dot{p}) \end{bmatrix}$$

$$g(w) = \begin{bmatrix} {}^{q}g(q) \\ {}^{p}g(p) \end{bmatrix} \tag{III.26}$$

The dynamics equation of the system is then

$$\tau = {}^{q}A(q) \cdot \ddot{q} + {}^{q}C(q, \dot{q}) \cdot \dot{q} + {}^{q}g(q) + J^{T}\left[{}^{p}A(p) \cdot \ddot{p} + {}^{p}C(p, \dot{p}) \cdot \dot{p} + {}^{p}g(p)\right]. \tag{III.27}$$

This model is called the *inverse dynamic model in the two spaces* since it uses the robot's state and output both in joint space and in operational space.

This formulation is not really suited for analysis since kinematic relations should be included in Eq. III.27 to reduce the dimension of the manipulator's state to $2n$ instead of $4n$. It is however a very interesting formulation when efficient implementation is required, for example in model-based control laws.

III.7 Efficient Computation of the Inverse Dynamics Model.

III.7.1 Intuitive Justification.

Considering Eq. III.2, Eq. III.27 may be interpreted as the sum of two partial models of the manipulator. The robot inertia, or its kinetic energy, is split in the two spaces and a model is established for each.

The model in joint space only represents apart of the dynamics that is independent of the closed-chain structure of the manipulator. Considering the double-tree kinematic model from Section II.7, the dynamics of both *base- and upper- trees*, considered as disconnected from each other, are included in this part of the model. The reaction on the base of the base tree is supposed to be absorbed, but the reaction on the base of the upper-tree is a force/torque that must somehow be applied on the end–effector, as indicated by the 6–DOF abstract joint in Fig. 8. The operational space part of the model represents this force/torque that is actually mapped on the actuators using Eq. III.2. Note that the inertia of the links is also split into the two sub–models, according to a position–dependent ratio.

On the basis of this idea, a new algorithm to evaluate systematically the inverse dynamics model of any parallel manipulator is proposed in § III.7.4. It uses a slightly enhanced version of the Newton–Euler algorithm proposed by Luh, Walker and Paul to compute efficiently the inverse dynamics model of robots with open chains [Luh80b].

III.7.2 The Newton–Euler Algorithm for Open–Chain Manipulators.

This algorithm is described in most robotics textbooks [Asad86], [Feat87], [Domb88]. The formulation from the latter reference is reminded here.

The structure's kinematics is described by its Denavit–Hartenberg parameters (see § II.5.3). The i^th^ body is described by its mass m_i, its inertia matrix ${}^{i}\mathrm{I}$ with respect to the i^th^ frame and the position of its center of gravity ${}^{i}g$ in the same frame. The inputs are the structure's state $\{q, \dot{q}\}$ and its acceleration $\ddot{q}$.

The algorithm consists of two iterative stages.

Stage 1: Forward Iteration.

In a forward iteration along the tree's structure from the base to the end–effectors, the relative velocities and accelerations are composed to obtain the absolute acceleration of each body. Then, the inertia forces/torques to apply on each can be computed. The convention about the numbering of the bodies in § II.5.3 allows to evaluate successively the expressions in Eq. III.28 for $i = 1, 2, \ldots n$.

$$\begin{cases}
{}^{i,a_i}T = \begin{bmatrix} {}^{i,a_i}R & {}^{i,a_i}p \\ 0 & 1 \end{bmatrix} & \text{(i)} \\
{}^{i,a_i}\omega = {}^{i,a_i}R \cdot {}^{a_i,a_i}\omega & \text{(ii)} \\
{}^{i,i}\omega = {}^{i,a_i}\omega + \bar{\sigma}_j\,{}^{i}z\dot{q}_i & \text{(iii)} \\
{}^{i,i}\dot{\omega} = \left[{}^{i,a_i}R \cdot {}^{i,a_i}\dot{\omega}\right] + \bar{\sigma}_i\left[{}^{i}z\ddot{q}_i + {}^{i,a_i}\omega \times {}^{i}z\dot{q}_i\right] & \text{(iv)} \\
{}^{i}U = {}^{i,i}\dot{\omega} + {}^{i,i}\omega \cdot {}^{i,i}\omega & \text{(v)} \\
{}^{i,0}\ddot{p} = {}^{i,a_i}R \cdot \left[{}^{a_i,0}\ddot{p} + {}^{a_i}U\,{}^{i,a_i}p\right] + \sigma_i\left[{}^{i}z\ddot{q}_i + 2\,{}^{i,a_i}\omega \times {}^{i}z\dot{q}_i\right] & \text{(vi)} \\
F_j = m_j\left[{}^{i,0}\ddot{p} + {}^{i}U \cdot {}^{i}g\right] & \text{(vii)} \\
N_i = {}^{i}I \cdot {}^{i,i}\dot{\omega} + {}^{i,i}\omega \times \left[{}^{i}I \cdot {}^{i,i}\omega\right] & \text{(viii)}
\end{cases} \qquad \text{(III.28)}$$

The initial conditions ${}^{0,0}\omega$, ${}^{0,0}\dot{\omega}$, ${}^{0,0}\ddot{p}$ correspond to the base's absolute acceleration and velocity in rotation. With ${}^{0,0}\ddot{p} = \{0, 0, 9.81\}^T$ for example, the effect of Earth's gravity is computed.

Stage 2 : Backward Iteration.

A backward iteration from the end-effectors to the base sums the torques and forces transmitted by the connected bodies and projects them on the corresponding joint. *The expressions in Eq. III.29 are* successively evaluated for $i = n, n-1, \ldots, 1$.

$$\begin{cases} {}^{i,i}f = F_i + \sum\limits_{k|a_k = i} {}^{k,a_k}f & \text{(i)} \\ {}^{i,a_i}f = {}^{i,a_i}R \cdot {}^{i,i}f & \text{(ii)} \\ {}^{i,i}\eta = N_i + \left[m_i {}^{i}g \times {}^{i,0}\ddot{p}\right] + \sum\limits_{k|a_k = i} \left[{}^{k,a_k}R \cdot {}^{k,k}\eta + {}^{k,a_k}p \times {}^{k,a_k}f\right] & \text{(iii)} \\ \tau_i = \left[\sigma_i {}^{i,i}f + \bar{\sigma}_i {}^{i,i}\eta\right]^{T\,i}z & \text{(iv)} \end{cases} \quad . \qquad \text{(III.29)}$$

The output is the vector τ of the torques/forces that are to be applied on each joint.

Complexity and Efficiency Comparison with Respect to Lagrange.

From the complexity point of view, a straightforward evaluation of Eq. III.15 is $O(n^3)$ and Luh, Walker and Paul's algorithm is $O(n)$. Lagrange and Newton–Euler formalisms were therefore first considered as complementary approaches to manipulators' dynamics (see Section III.2), the former allowing analysis, the latter more suited for implementation.

The raw Newton–Euler algorithm was reported to require $238n - 33$ floating point elementary operations (FLOP) in the case of a general arm [Luh80b]. However, when redundant computations are eliminated, only *$173n - 244$ FLOP are needed [Domb88]. For a given arm, this burden is again reduced* by about 30%. Software packages such as SYMORO allow to generate automatically the minimal code to implement Newton–Euler's algorithm for any open-chain robot [Domb88].

Later, Lagrange and Newton–Euler approaches were later shown to be formally equivalent [Silv82]. When redundant computations were removed from Eq. III.15 [Bala88], efficient implementations of Lagrange dynamics where proposed that seem to require even fewer computations than a Newton–Euler algorithm [Li88], [Baha91]. The main difference that remains is the structure of the algorithms:

- Because of its iterative structure, the Newton–Euler algorithm can efficiently be implemented on *pipelined* or *vector processors* such as *digital signal processors* (DSP). *Multiprocessor* control units have also been used after some reformulation of the algorithm [Chen88], [LeeG86].

- The evaluation of the raw Lagrange model from Eq. III.15 can greatly benefit from n processors working in parallel since the complexity on each is reduced to $O(n^2)$ [Zoma92]. However, this matrix structure vanishes when redundant computations are removed.

III.7.3 Extension.

The first Stage of the Newton–Euler algorithm produces as a side–effect the absolute translation acceleration ${}^{i,0}\ddot{p}$ (see Eq. III.28.vi) and relative angular acceleration ${}^{i,i}\dot{\omega}$ (see Eq. III.28.iv) of each body. The absolute positions, velocities and angular accelerations require two more expressions (75 FLOP) to be computed at each step of the first stage:

$$\begin{cases} {}^{i,0}T = {}^{a_i,0}T \cdot {}^{i,a_i}T = \begin{bmatrix} {}^{i,0}R & {}^{i,0}p \\ 0 & 1 \end{bmatrix} & \text{(i)} \\ {}^{i,0}\dot{p} = {}^{a_i,0}\dot{p} + {}^{a_i,a_i}\omega \times {}^{i,a_i}p & \text{(ii)} \end{cases} \quad . \tag{III.30}$$

The absolute positions, speeds and accelerations of the end–effector(s) can then be obtained by evaluating Eq. III.28ii,ivv,vi and Eq. III.30 for $i = n+1, n+2, \ldots, n+n_e$. In other words, the forward kinematics relations Eq. II.10, Eq. II.15 and Eq. II.29 are evaluated simultaneously with the inverse dynamics model at the expense of 75 FLOP per end–effector. Remembering that the matrix ${}^{i,a_i}R$ contains the direction cosines of the end–effector's frame with respect to the frame attached to the last body in the chain, one can consider that, in most practical cases, ${}^{i,a_i}R$ contains only three nonzero elements that are either 1 or –1. The extra–cost per end–effector is then only 30 FLOP.

When the end–effectors have interactions with their environment, external forces and torques can then be added to the corresponding F_i and N_i. Stage two is then evaluated for $i = n+n_e, n+n_e-1, \ldots, 1$, requiring $15n_e$ FLOP more than the standard algorithm. This extended algorithm therefore requires about $284n + 45n_e - 244$ FLOP for a general tree robot.

III.7.4 A New Algorithm for Parallel Manipulators.

The manipulator's kinematics is parametrized as described in Section II.7. The dynamics parameters of the bodies in both trees are defined as in § III.7.2.

The inputs to the algorithm are the manipulator's state and acceleration in both joint and operational spaces $\{q, \dot{q}, \ddot{q}, p, \dot{p}, \ddot{p}\}$, which must satisfy the constraint equations, that is the kinematic relations between the two spaces. Moreover, the Jacobian matrix is also required. The output is the vector τ of the torques/forces to apply on each actuated joint.

Stage 1: Forward Newton–Euler Iteration on the Trees.

The forward iteration of the Newton–Euler algorithm from § III.7.2 is applied on the base– and upper–trees. The absolute acceleration of the twelve connection points ${}^{b}P_i$ and ${}^{u}P_i$ is obtained at the same time.

- Compute Eq. III.28 and Eq. III.30 for the base–tree for $i = 1\ldots\left({}^{b}n + 6\right)$, with ${}^{0,0}\omega = 0$, ${}^{0,0}\dot{\omega} = 0$, ${}^{0,0}\dot{p} = 0$ and ${}^{0,0}\ddot{p}$ equal to the opposite of the acceleration due to gravity. The positions ${}^{b}P_i = {}^{n+i,0}p$, speeds ${}^{b}\dot{P}_i = {}^{n+i,0}\dot{p}$ and accelerations ${}^{b}\ddot{P}_i = {}^{n+i,0}\ddot{p}$ of the base–tree connection points are obtained as well as their rotation speed ${}^{b}\Omega_i = {}^{n+i,0}R \cdot {}^{n+i,a_{n+i}}\omega$ and accelerations ${}^{b}\dot{\Omega}_i = {}^{n+i,0}R \cdot {}^{n+i,a_{n+i}}\dot{\omega}$.
- Depending on the chosen representation of the orientations (see § II.3.4), extract ${}^{0,0}\omega$ from $\dot{p}$, and ${}^{0,0}\dot{\omega}$ *from* $\ddot{p}$. *Initialize* ${}^{0,0}\ddot{p}$ *as the translation acceleration in* $\ddot{p}$ *minus the gravity acceleration.*
- Compute Eq. III.28 and Eq. III.30 for the upper–tree for $i = 1\ldots\left({}^{u}n + 6\right)$. This part is often trivial since ${}^{u}n = 0$ in most practical cases (see Table 1). The positions, velocities and accelerations of the upper–tree connection points are obtained as for the base tree.
- Since the root of the upper–tree is a free body holding the end–effector with a given mass m_0 and inertia ${}^{0}I$, compute the torque/force on it with Eq. III.28.vi and Eq. III.28.vii, $i = 0$.

Stage 2: Links solution.

Since the positions, velocities and accelerations at both edges of each of the 6 links are known, the *forces to apply at the connection points of the real links are computed.*

For the sake of clarity, the ${}^{b}P_i{}^{u}P_i$ axis of the ith link is supposed to be an axis of symmetry with the center of gravity in its middle. These hypotheses are fulfilled in most practical cases and greatly reduce the overall complexity of the obtained model. The dynamics parameters of the ith link are therefore its mass ${}^{l}m_i$, its inertia about the ${}^{b}P_i{}^{u}P_i$ axis ${}^{\parallel}i_i$ and ${}^{\perp}i_i$ about any orthogonal axis crossing the center of gravity.

A useful vector relation is first reminded. Any vector w can be decomposed as

$$w = {}^{v\parallel}w + \left(v \times {}^{v\perp}w\right), \qquad \text{(III.31)}$$

where ${}^{v\parallel}w = (w^T \cdot v)\,v$ is parallel to the unit vector v and ${}^{v\perp}w = w \times v$ orthogonal to it (see Fig. 10).

Consider the ith link. Its length l_i is constant if the link is rigid or $l_i = q_j$ where j is the index of the corresponding joint variable in vector q.

- First, a unit vector along the link is defined by ${}^{i}v = \dfrac{{}^{u}P_i - {}^{b}P_i}{|l_i|}$.

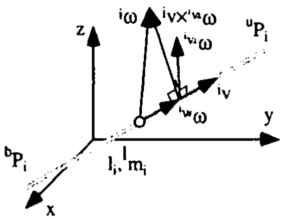


Fig.10 : Link with Colinear and Orthogonal Decomposition of Velocities.

- Since ${}^{u}\dot{P}_i - {}^{b}\dot{P}_i = \dot{q}_j{}^{i}v + |l_i|{}^{i}v \times {}^{i}\omega$, where ${}^{i}\omega$ is the rotation about the center of gravity of the link (see Fig. 10) and q_j is the joint variable associated to the link in the case of an actuated link :

$$^{{}^{i}v\perp}\omega = {}^{i}\omega \times {}^{i}v = \frac{\dot{q}_j{}^{i}v - \left({}^{u}\dot{P}_i - {}^{b}\dot{P}_i\right)}{|l_i|}. \tag{III.32}$$

- The rotation speed around ${}^{i}v$ is imposed by the universal joint:

$$^{{}^{i}v''}\omega = \begin{cases} \left({}^{b}\Omega_j{}^{T} \cdot {}^{i}v\right){}^{i}v, l_i > 0 \\ \left({}^{u}\Omega_j{}^{T} \cdot {}^{i}v\right){}^{i}v, l_i < 0 \end{cases}. \tag{III.33}$$

- The same constraint on the accelerations gives the torques ${}^{b,i}\eta$ and ${}^{u,i}\eta$ to apply at both edges of the links:

$$\begin{cases} {}^{b,i}\eta = \left({}^{b}\dot{\Omega}_j{}^{T} \cdot {}^{i}v\right)^{\prime\prime} i_i, {}^{u,i}\eta = 0, {}^{i}l_i > 0 \\ {}^{u,i}\eta = \left({}^{u}\dot{\Omega}_j{}^{T} \cdot {}^{i}v\right)^{\prime\prime} i_i, {}^{b,i}\eta = 0, {}^{i}l_i < 0 \end{cases} \tag{III.34}$$

- If the link is actuated, $\tau_j = {}^{l}m_i\ddot{q}_j$.
- The orthogonal rotation acceleration around the center of gravity can be obtained from the relation

$${}^{u}\ddot{P}_i - {}^{b}\ddot{P}_i = {}^{i}v\left[\ddot{q}_j - |l_i|\left\|{}^{{}^{i}v\perp}\omega\right\|^2\right] + {}^{i}v \times \left[|l_i|{}^{{}^{i}v\perp}\dot{\omega} - |l_i|\left(2\dot{q}_j - \left({}^{{}^{i}v''}\omega \cdot {}^{i}v\right)\right){}^{{}^{i}v\perp}\omega\right]: \tag{III.35}$$

- Finally, the forces ${}^{b,i}f$ and ${}^{b,i}f$ applied at both edges of the link must satisfy Newton equations:

$$\begin{cases} {}^{b,i}f + {}^{u,i}f = \dfrac{{}^{l}m_i}{2}\left[{}^{u}\ddot{P}_i + {}^{b}\ddot{P}_j\right] \\ |l_i|\,{}^{i}v \times \left[{}^{i}v \times \left[{}^{u,i}f - {}^{b,i}f\right]\right] = {}^{\perp}i_i\left[\dfrac{\left({}^{u}\ddot{P}_i - {}^{b}\ddot{P}_i\right)}{|l_i|} - {}^{i}v\left[\dfrac{\ddot{q}_j}{|l_i|} - \left\|{}^{{}^{i}v\perp}\omega\right\|^2\right] - \left(2\dot{q}_j - \left({}^{{}^{i}v\sim}\omega^T \cdot {}^{i}v\right)\right){}^{i}v \times {}^{{}^{i}v\perp}\omega\right] \end{cases}$$

(III.36)

Eq. III.36 is a linear system of six equations for the six unknown components of ${}^{b,i}f$ and ${}^{u,i}f$. However, since the forces on both edges act on five degrees-of-freedom of the link, the system is *under-determined. One may consider adding a constraint equation such as* ${}^{b,i}f \cdot {}^{i}v - {}^{u,i}f \cdot {}^{i}v = 0$ to prevent internal stress to be created in the link. However, this constraint will anyway be satisfied in Stage 4, which maps the artificial forces onto the actuators. It is therefore sufficient to fix any single component of ${}^{b,i}f$ or ${}^{u,i}f$ to an arbitrary value and to solve Eq. III.36 for the five remaining unknowns.

Stage 3: Backward Newton-Euler Iteration on the Trees.

The backward iteration of the Newton-Euler algorithm from § III.7.2 is applied on the base- and upper-trees to obtain the torques/forces on the actuated joints in both and the torque/force on the end-effector.

- Add the forces and torques required to move the links to the connection points on the base-tree

$$\begin{cases} F_{n+i} \leftarrow F_{n+i} + {}^{b,i}f \\ N_{n+i} \leftarrow N_{n+i} + {}^{b,i}\eta \end{cases} \quad \text{(III.37)}$$

- Compute Eq. III.29 for the base- tree.
- Add the forces and torques required to move the links to the connection points on the upper-tree

$$\begin{cases} F_{n+i} \leftarrow F_{n+i} + {}^{u,i}f \\ N_{n+i} \leftarrow N_{n+i} + {}^{u,i}\eta \end{cases} \quad \text{(III.38)}$$

- Compute Eq. III.29 for the upper- tree.
- Evaluate Eq. III.29 for $i = 0$ to obtain the reaction force ${}^{0,0}f$ and torque ${}^{i,i}n$ on the end-effector. According to the representation of the orientations, these force and torque vectors can be combined in a *n-DOF torque/force vector in operational space* ${}^{u,0}F$. An external force/torque exerted by the environment can here be added if required.

Stage 4: Mapping of the Forces on the Upper Tree's Root onto the Joints.

- Finally, the forces/torques in operational space to apply on the end–effector are mapped onto the joint space with the fundamental relation Eq. III.2:

$$\tau = \begin{bmatrix} {}^{b}\tau \\ {}^{u}\tau \\ {}^{l}\tau \end{bmatrix} + J^{T} \cdot {}^{u,0}F. \tag{III.39}$$

III.7.5 Complexity, Optimization and Structure of the Algorithm.

Complexity.

Stages 1 and 3 compute the extended Newton–Euler algorithm from § III.7.3 once for the upper–tree and once for the base–tree. Stage 2 requires 119 FLOP for a general, actuated link and 80 FLOP for a rigid link. Stage 4 requires $2n^2$ FLOP. For $n \le 6$, this is negligible compared to the other stages.

The total number of computations for a general parallel manipulator is therefore $2n^2 + 284\left({}^{b}n + {}^{u}n\right) + 39\,{}^{l}n + 959$ FLOP. For a Stewart platform(${}^{b}n = 0,\ {}^{u}n = 0,\ {}^{l}n = 6$), this is about the half of the cost of the standard Newton–Euler algorithm for a general 6–DOF serial arm. For the Hexa, which is structurally very similar (${}^{b}n = 6,\ {}^{u}n = 0,\ {}^{l}n = 0$), the cost is surprisingly about 50% higher than for a serial arm.

From the purely dynamics point of view, one can therefore consider that the complexity of the inverse dynamics model evaluation for a n–DOF parallel manipulator has the same order of magnitude as for a n–DOF robot with open chains.

Comparison with Lagrange and Natural Coordinates.

Since Lagrange and Newton–Euler algorithms were shown to be equivalent for open–chain robots (see § III.7.2), Stage 2 is the most apparent difference between these approaches. The forces at the connection points are explicitly computed with the Newton–Euler algorithm, while a Lagrange approach would implicitly only consider its energy. In this respect, the Newton–Euler algorithm is closer to a natural coordinates approach where the absolute positions, velocities and accelerations of each body are given in a unique reference frame where all the forces, working or not, are also expressed. For this reason, it is believed that many redundant or useless computations can be removed from the raw formulation of the proposed algorithm.

Importance of Kinematics Computations.

The cost of the kinematics transformations that are required to obtain the robot's state in the two spaces as well as the Jacobian matrix should also be considered. The proposed dynamics algorithm contains many expressions that are redundant with kinematics computations. Still, the main problem remains to obtain the Jacobian matrix that solves the closure constraints. As mentioned in § II.9.4, the only practical solution is to invert the inverse Jacobian matrix, which is very time consuming. This is probably why many authors consider parallel manipulators as much more involved than serial arms, especially if they are not aware of the key importance of the **J** matrix.

A Parallel Structure.

It is important to note that he proposed algorithm has an inherently parallel structure instead of the iterative structure of the original Newton-Euler algorithm : Stage 2 can be computed for each link independently, and this parallelism is propagated along both upper- and base- trees as long as the corresponding sub-chains are not connected. How to exploit this structure and how to combine the kinematics computations with it in a multiprocessor control unit is discussed in detail in Section IV.7.

III.8 A General Framework: the Operational Space Formulation.

III.8.1 Projection of the Model in the Two Spaces onto Operational Space only.

Since no closed form of the direct kinematic relation f exists, it is impossible to obtain a closed form inverse dynamics model in joint space only. In contrast, introducing the inverse kinematics transformations from Eq. II.11, Eq. II.17 and Eq. II.31 in Eq. III.27 gives an inverse dynamics model in task space only:

$$\begin{aligned} \tau = {}& {}^{q}A'(p) \cdot \left[J^{-1}(p) \cdot \ddot{p} + \dot{J}^{-1}(p, \dot{p}) \cdot \dot{p}\right] + \\ & + {}^{q}C'(p, \dot{p}) \cdot J^{-1}(p) \cdot \dot{p} + {}^{q}g'(p) + \\ & + J^{T}\left[{}^{p}A(p) \cdot \ddot{p} + {}^{p}C(p, \dot{p}) \cdot \dot{p} + {}^{p}g(p)\right] \end{aligned} \tag{III.40}$$

where a prime indicates a parameter substitution $q \rightarrow f^{-1}(p)$ to prevent any confusion with the notations introduced in Section III.4.

Grouping terms together and dropping all the p and $\dot{p}$ parameters to clarify yields:

$$\tau = J^{T} \cdot \upsilon , \tag{III.41}$$

where

$$\upsilon = \left[{}^{p}A + J^{-T\,q}A' J^{-1}\right] \cdot \ddot{p} + \left[J^{-T}\left({}^{q}A' \dot{J}^{-1} + {}^{q}C' J^{-1}\right) + {}^{p}C\right] \cdot \dot{p} + \left[J^{-T\,q}g' + {}^{p}g\right] \tag{III.42}$$

Combining Eq. III.42 and Eq. III.41 gives the *inverse dynamics model in operational space*

$$\tau = M(p) \cdot \ddot{p} + D(p, \dot{p}) \cdot \dot{p} + h(p) , \tag{III.43}$$

with

$$\begin{aligned} M &= J^{T\,p}A + {}^{q}A' J^{-1} , \\ D &= {}^{q}A' \dot{J}^{-1} + {}^{q}C' J^{-1} + J^{T\,p}C , \\ h &= {}^{q}g' + J^{T\,p}g . \end{aligned} \tag{III.44}$$

For manipulators with open chains only, Eq. III.43 is equivalent to Eq. III.17. Contrarily to the formulation in joint space from Eq. III.15 that is applicable to serial arms and tree–robots only, this formulation is very general and should be used in future research:

- *to obtain really general theoretical results that are applicable to all manipulators,*
- whenever contact forces are to be included in the model,
- to develop universal controllers that may drive a wide class of robots.

III.8.2 Properties.

A problem with the formulation in operational space is that Eq. III.43 is not well suited for analysis since the pseudo–inertia matrix $M(p)$ and the pseudo–quadratic forces matrix $D(p, \dot{p})$ do not possess the properties of real inertia matrix and quadratic forces matrix discussed in Section III.5. However, one can consider that Eq. III.42 contains the true dynamic model of the mechanism while Eq. III.41 simply projects the input forces υ that should be exerted in operational space onto the actuated joints.

Analysis can therefore be performed by studying Eq. III.42 alone, with an inertia matrix, a quadratic forces matrix and a gravity vector defined as follows

$$^{o}A = {}^{p}A + J^{-T\,q}A' J^{-1}, \qquad \text{(III.45)}$$

$$^{o}C = J^{-T\,q}A' \dot{J}^{-1} + J^{-T\,q}C' J^{-1} + {}^{p}C, \qquad \text{(III.46)}$$

$$^{o}g = J^{-T\,q}g' + {}^{p}g. \qquad \text{(III.47)}$$

Some important properties of the dynamics of manipulators are obtained below from this formulation. The proofs rely on three trivial properties from linear algebra.

Property 1: if S is symmetric, then $T^T \cdot S \cdot T$ is symmetric, $\forall T \in \mathcal{R}^{n \times n}$.

Proof: $S = S^T \Rightarrow T^T \cdot S \cdot T = [T^T \cdot S^T \cdot T]^T = [T^T \cdot S \cdot T]^T$.

Property 2: if S is skew-symmetric, then $T^T \cdot S \cdot T$ is skew-symmetric, $\forall T \in \mathcal{R}^{n \times n}$.

Proof: $S = -S^T \Rightarrow T^T \cdot S \cdot T = [T^T \cdot S^T \cdot T]^T = -[T^T \cdot S \cdot T]^T$.

Property 3: *if S is positive definite, then $T^T \cdot S \cdot T$ is positive definite, $\forall T \in \mathcal{R}^{n \times n}$ regular.*

Proof: Since S is positive definite, $x^T \cdot T^T \cdot S \cdot T \cdot x > 0 \Rightarrow T \cdot x \neq 0$.
Since T is regular, $T \cdot x = 0 \Leftrightarrow x = 0$.
Therefore, $\forall x \in \mathcal{R}^n, x \neq 0$, $x^T \cdot T^T \cdot S \cdot T \cdot x > 0$, i.e. $T^T \cdot S \cdot T$ is positive definite.

Symmetry of the Inertia Matrix in Operational Space.

Theorem 1: $^{o}A = {}^{p}A + J^{-T\,q}A' J^{-1}$ is symmetric $\forall p \in O \,\big|\, |J^{-1}(p)| \neq 0$

Proof: Since $A(w)$ is symmetric, ^{p}A and $^{q}A'$ are symmetric by their definition in Eq. III.26. Property 1 implies that $J^{-T\,q}A' J^{-1}$ is symmetric and ^{o}A is therefore symmetric since it is the sum of two symmetric matrices.

Positive Definiteness of the Inertia Matrix in Operational Space.

Theorem 2: $^{o}A = {}^{p}A + J^{-T\,q}A' J^{-1}$ is positive definite (PD) for p such that $J^{-1}(p)$ is regular.

Proof: Since $A(w)$ is PD, ^{p}A and $^{q}A'$ are PD by their definition in Eq. III.26. Since $^{q}A'$ is PD, $J^{-T\,q}A' J^{-1}$ is also PD by Property 3. ^{o}A being the sum of PD matrices is PD itself.

Passivity.

Theorem 3: $^{o}A - 2^{o}C$ is skew-symmetric (SS) $\forall p \in \mathcal{O} \,|\, |J^{-1}(p)| \neq 0$.

Proof: From Eq. III.45 and Eq. III.46,

$$^{o}A - 2^{o}C = \left[{}^{p}\dot{A} - 2^{p}C\right] + J^{-T}\left[{}^{q}\dot{A}' - 2^{q}C'\right]J^{-1} + \left[\dot{J}^{-T\,q}A'J^{-1} - J^{-T\,q}A'\dot{J}^{-1}\right]$$

$^{p}\dot{A} - 2^{p}C$ and $^{q}\dot{A}' - 2^{q}C'$ are SS because of the definition of ^{p}C and $^{q}C'$ from Eq. III.11, Eq. III.13 and Eq. III.26. Hence, using Property 2, $J^{-T}\left({}^{q}\dot{A}' - 2^{q}C'\right)J^{-1}$ is SS. Finally, since $\dot{J}^{-T\,q}A'J^{-1} = \left[{}^{q}A'\dot{J}^{-1}\right]^{T}J^{-1} = \left[J^{-T\,q}A'\dot{J}^{-1}\right]^{T}$, $\dot{J}^{-T\,q}A'J^{-1} - J^{-T\,q}A'J^{-1}$ is also SS and $^{o}A - 2^{o}C$ being the sum of SS matrices is SS.

Theorem 4: The operational space dynamics Eq. III.42 of a rigid manipulator define a passive mapping $\upsilon \rightarrow \dot{p}$, i.e. $\forall t_f \in \mathcal{R}^{+}, \exists \beta > 0 \,|\, \int_{0}^{t_f} (\upsilon^{T} \cdot \dot{p})\, dt \geq \beta$ in any subspace of $\mathcal{O}$ where inverse singularities are excluded.

Proof: This proof is derived from the one given in [Orte89] for serial arms using joint space coordinates only.

The *Hamiltonian* $e_H(t) = \frac{1}{2}\left(\dot{p}(t)^{T} \cdot {}^{o}A(p(t)) \cdot \dot{p}(t)\right) + e'_P(p(t))$ of the manipulator at a given time t is defined as the sum of the kinetic and potential energies expressed in operational space. Differentiating it with respect to time and dropping parameters for clarity yields $\frac{de_H}{dt} = \dot{p}^{T} \cdot {}^{o}A \cdot \ddot{p} + \frac{1}{2}\left[\dot{p}^{T} \cdot {}^{o}\dot{A} \cdot \dot{p}\right] + \frac{\partial}{\partial p}e'_P \cdot \dot{p}$. Introducing the dynamic equation obtained from Lagrangian mechanics gives

$\frac{de_H}{dt} = \dot{p}^{T} \cdot \upsilon + \frac{1}{2}\dot{p}^{T} \cdot \left[{}^{o}A - 2^{o}C\right] \cdot \dot{p}$. The second term is zero because of Theorem 3, hence $\frac{de_H}{dt} = \dot{p}^{T} \cdot \upsilon$. Consequently,

$\int_{0}^{t_f} (\upsilon^{T} \cdot \dot{p})\, dt = e_H(t_f) - e_H(0) \geq -e_H(0)$ since e_H represents an energy and is therefore always non negative.

Passivity in Joint Space.

Theorem 5: The dynamics of a rigid manipulator define a passive mapping $\tau \rightarrow \dot{q}$, i.e.

$$\forall t_f \in \mathcal{R}^+, \exists \beta > 0 \mid \int_0^{t_f} (\tau^T \cdot \dot{q})\, dt \geq \beta$$ in the subspace of $\mathcal{O}$ where singularities of both the Jacobian matrix and its inverse are excluded.

Proof: Eq. II.15 and Eq. III.41 give $\tau^T \cdot \dot{q} = \upsilon^T \cdot J \cdot J^{-1} \cdot \dot{p} = \upsilon^T \cdot \dot{p}$ if J and J^{-1} are both regular. Since by Theorem 4 $\upsilon \rightarrow \dot{p}$ is passive, $\tau \rightarrow \dot{q}$ is also passive.

III.8.3 Is Passivity a Local Property?

Physically, a dynamic system is said passive when its internal energy fades away when no external energy is supplied. This is obviously a global property of all Hamiltonian systems, which include articulated mechanisms. It should therefore be clear that the manipulator itself is globally a passive system, while Theorem 5 above only show that the specific mapping $\tau \rightarrow \dot{q}$ is passive only in the regular workspace of the robot. Unfortunately, applications to control rely on the existence of passive mappings, not on the global property. However, as long as the robot is operated in its restricted workspace, this limitation has no practical consequences.

III.8.4 A Critical Discussion on the Inertia of Manipulators.

Variation Range.

Some elements of ${}^qA(q)$ can have very small values at certain configurations, for example when a body is outstretched along a revolute joint's axis. The corresponding elements of ${}^qA(q)$ may therefore vary by one or even two orders of magnitude for a typical serial manipulator. This situation is unlikely to happen in a parallel manipulator where the structure prevents to reach such extreme configurations. ${}^qA$ varies much less. Moreover, since the idea behind parallel manipulators is to lighten the carrier structure with respect to the end-effector, it is reasonable to assume that ${}^qA$ is not larger than ${}^pA$. As a consequence, ${}^oA$ can be considered as almost constant and equal to ${}^pA$ in the robot's restricted workspace. The main cause of variations of ${}^oA$ in a parallel manipulator is therefore the payload, which plays a much smaller role on a serial arm.

Smoothness and Boundedness.

Eq. III.9 reveals that $A(w)$ is obtained by projecting the directions of possible displacements of the center of gravity of each body in the structure onto joint space. Considering the double-tree model, ${}^qA$ contains sums and products of the n local Jacobians that correspond to the center of gravity of each of the bodies in the upper-tree and base-tree. Since the proof of the smoothness and boundedness of the Jacobian of open chain mechanisms(§ II.9.1) also holds for the local Jacobians, ${}^{q}A(q)$ is a *smooth* function of q. A more formal proof of these properties can be found in [Spon89]. ${}^{q}A(q)$ is also *bounded* for manipulators with bounded prismatic joints' strokes.

A similar development can be made for ${}^{p}A$, which is also a smooth and bounded function of p. However, ${}^{o}A(p)$ is formally neither smooth nor bounded for manipulators since the inverse Jacobian has none of these properties (see § II.9.1). From a practical point of view, it seems nevertheless reasonable to assume that ${}^{o}A(p)$ is smooth and bounded in the robot's restricted workspace.

A Critical Comparison.

When the pseudo–inertia matrix $M(p) = J^T \cdot {}^{o}A$ is considered, an upper bound can clearly be found for serial arms, the lower bound of some elements being close to zero. For parallel manipulators, a lower bound exists while an upper bound depends on the definition of the robot's restricted workspace since some elements of $M(p)$ are infinite at the inverse singularities.

It is therefore not true to state that parallel robots offer better performances than serial arms because of their lower inertia. The situation is exactly the same as for the stiffness examined in § III.3.1: the advantages are restricted to a subspace of the workspace that should be very carefully defined.

III.8.5 A Note on Discrete–Time Models.

All the theory on manipulator dynamics has been developed in the continuous–time domain. Since controllers are implemented on digital processors, research has tried to obtain discrete–time models in the past. Neuman and Tourassis proposed a model that respects the principle of conservation of the total energy [Neum85], [Tour85]. However, this is accomplished by sacrificing other properties such as the existence of passive mappings that are fundamental in control. Actually, the quest for discrete–time robot models ended when digital processors became powerful enough to reach high sampling rates where the effect of sampling becomes negligible in the closed–loop bandwidth.

III.9 Conclusions.

Parallel manipulators' dynamic models can only be obtained in closed form using operational space *state variables. Using Lagrangian mechanics, one can easily obtain such a model for a given manipulator* when an inverse kinematics model is available. The equivalence of the cancellation of Lagrange multipliers with the computation of the Jacobian is shown. An inverse dynamics model "in the two spaces" is developed that can be efficiently implemented when fast evaluation of the inverse dynamics is required.

Dynamics analysis of parallel manipulators is carried out and important properties such as the passivity of the $\upsilon \rightarrow \dot{p}$ mapping are shown. With some weak restrictions, the passivity of $\tau \rightarrow \dot{q}$ is also proved, allowing advanced control methods developed for serial arms to be applied to parallel robots.

Finally, some critical considerations about fundamental properties of parallel manipulators are made. It is showed that the alleged higher stiffness and lower inertia of parallel manipulators have no justification if the robot's workspace is not correctly defined.

Chapter IV

Control.

IV.1 Introduction.

The aim of the first four Sections is to provide a survey of the existing approaches to the control of serial robots. As shown in Chapter III, the dynamics of parallel manipulators operated in their regular workspace have the same fundamental properties as serial robots. Hence, the full set of strategies available to control serial arms can theoretically be applied to parallel manipulators.

First, the classic control strategies based on linear theory and their limitations are discussed in Section IV.2. The importance of additional dynamics such as frictions, actuator dynamics and joint elasticity is also examined.

Then, the concept of feedback linearization is introduced in Section IV.3, where the computed-torque scheme is given. Control schemes in operational space that are equivalent to schemes in joint space are also introduced. Experimental results obtained with various setups are reported.

Some adaptive techniques are then reviewed in Section IV.4 to be complete. Finally, different robust control schemes are detailed in Section IV.5. Many of the existing control laws contain both a model-based part and robust part. This tradeoff is discussed with respect to recent theoretical results and practical considerations.

The case of fast parallel manipulators is considered in Section IV.6. The tight connection between dynamics and kinematics is discussed and it is shown that a control scheme in operational space requires fewer computations than its equivalent in joint space. A model-based control scheme in operational space is therefore proposed. It uses the real-time evaluation of the "inverse dynamics model in the two spaces" developed in Chapter III.

Implementation issues are discussed in Section IV.7. It is shown that the algorithms involved in the proposed control scheme have an inherent parallel structure that can be exploited on a multiprocessor control unit. The efficiency of the implementation can be further improved by introducing a pipeline in the computations. An experimental Transputer-based control unit corresponding to this design is briefly described. Finally, the main contributions offered in this Chapter are reviewed in Section IV.8.

IV.2 Classical Control Strategies.

IV.2.1 Dynamics of Robots with Transmissions.

In most existing industrial serial arms, a transmission is introduced between each actuator and the corresponding articulated body of the manipulator. Such an actuated joint is illustrated in Fig. 11. For the sake of clarity, the i subscript is ommited in this paragraph.

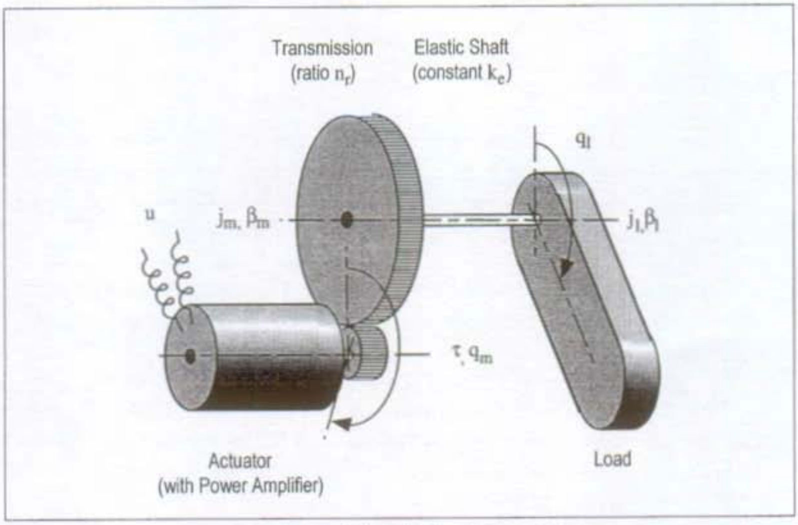


Fig.11 : Actuated Joint with Elastic Transmission.

Rigid Model.

Increasing the reduction ratio n_r has the advantage of lowering the influence of the load's parameters on the dynamics. For $n_r > 10$, coupling effects and quadratic forces become negligible with respect to other phenomena such as frictions that are amplified by the presence of the reduction. In this context, if gravity is considered as an external, unknown perturbation, the manipulator's dynamics equation from Eq. III.15 reduces to n independent single-input, single-output (*SISO*) linear equations.

The transfer function between the body's position q_l and the torque τ applied by the actuator is first given in the case of an ideal perfectly rigid transmission ($k_e \to \infty$) :

$$\frac{q_l(s)}{\tau(s)} = \frac{k_e n_r}{s\left(\left(j_m + \frac{j_l}{n_r^2}\right)s + \left(\beta_m + \frac{\beta_l}{n_r^2}\right)\right)} = \frac{\frac{k_e n_r}{A}}{s\left(s + \frac{\beta}{A}\right)}, \tag{IV.1}$$

where

$$A = j_m + \frac{j_l}{n_r^2} \tag{IV.2}$$

is the element on the diagonal of the inertia matrix that represents the approximately constant inertia driven by the actuator and

$$\beta = \beta_m + \frac{\beta_l}{n_r^2} \tag{IV.3}$$

is the kinetic friction coefficient of the whole.

This transfer function has a pole in zero (integrator) and another one in $s = -\frac{\beta}{A}$ which is also very close to the origin when kinetic friction in the system is minimized (see Fig. 12).

Drawbacks of Reductions.

Several undesirable phenomena are created or amplified by the presence of reductions:

- Stick-slip friction, which is very difficult to model accurately. Moreover, the stick-slip phenomenon is associated with elastic deformations of the material where potential energy can be stored and released according to very local properties of the material's surface. As a consequence, the passivity property may not be preserved when this happens.
- Backlash is often met when trying to reduce stick-slip in transmissions. It is highly undesirable for the same reasons.
- Kinetic friction is enlarged. While this poses no fundamental problem, the manipulator is more damped and more power is dissipated in heat in the reductions.
- Elasticity is boosted. One may therefore be tempted to consider explicitly elasticity in the model.

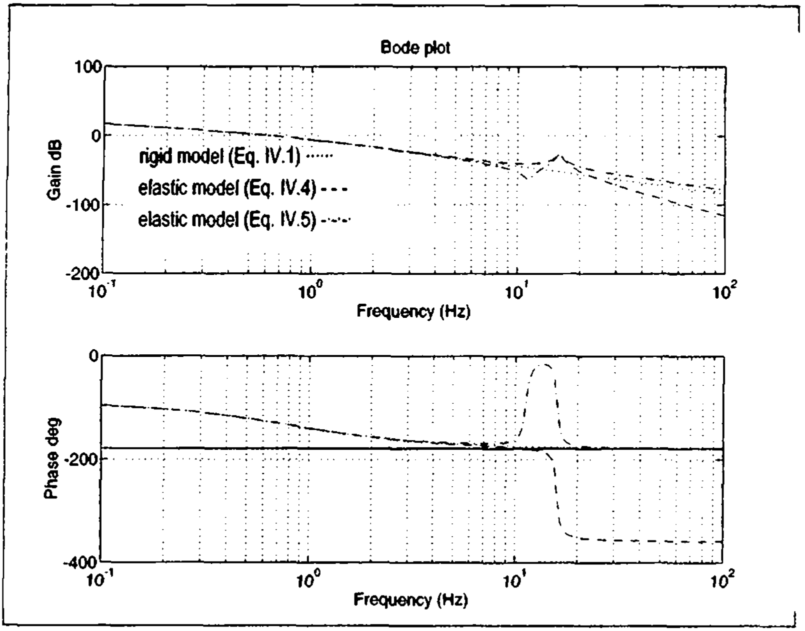


Fig.12 : Typical Open Loop Frequency Responses of a Serial Arm Joint.
(Elbow joint of a Puma 560 [Tam93])

Elastic Model.

When elasticity is considered, Eq. IV.1 is replaced by a fourth-order model [Spon89]:

$$\frac{q_l(s)}{\tau(s)} = \frac{k_e n_r}{s\left(\frac{j_m j_l}{k_e}s^3 + \left(\frac{j_m \beta_l + j_l \beta_m}{k_e n_r}\right)s^2 + \left(j_m + \frac{j_l}{n_r^2} + \frac{\beta_l \beta_m}{k_e}\right)s + \left(\beta_m + \frac{\beta_l}{n_r^2}\right)\right)}. \tag{IV.4}$$

Note that the transfer function between the actuator's position q_m and τ_m has the same poles, but two stable zeros bring the relative degree back to two:

$$\frac{q_m(s)}{\tau(s)} = \frac{\frac{j_l}{k_e}s^2 + \frac{\beta_l}{k_e}s + 1}{s\left(\frac{j_m j_l}{k_e}s^3 + \left(\frac{j_m \beta_l + j_l \beta_m}{k_e n_r}\right)s^2 + \left(j_m + \frac{j_l}{n_r^2} + \frac{\beta_l \beta_m}{k_e}\right)s + \left(\beta_m + \frac{\beta_l}{n_r^2}\right)\right)} \tag{IV.5}$$

Two complex conjugates poles appear that reveal the resonant mode associated to the elasticity of the transmission. This mode is generally badly damped as a consequence of the designer's effort to get rid of frictions. The corresponding *natural frequency* heavily depends on the considered robot joint. For the three firsts joints of a typical serial industrial manipulator like the Puma 560, they are in the 5-15 Hz range [Tam93]. Natural frequencies higher than 100 Hz are common on small parallel manipulators h as the Delta or in *wrists*.

Actuator Dynamics.

A model of the actuator itself should be appended to the above purely mechanical descriptions to obtain the complete model of the process in Fig. 11.

Each of the robot's actuators produces a torque/force τ_i that is related to the control signal u_i applied at the actuator's power amplifier. In modern electrical drives, the latter features a current control loop to achieve torque/force tracking, i.e. $\tau_i \sim u_i$. Usually, the presence of switching power devices such as thransistors naturally lead to high gain control strategies. The closed-loop bandwidth is then mostly limited by the motor's inductance only. From the user's point of view, the actuator's behavior can be approximated by a first order transfer function

$$H_i(s) = \frac{\tau_i(s)}{u_i(s)} = \frac{k_{t,i}}{1 + t_{m,i}s}, \tag{IV.6}$$

where $k_{t,i}$ is the torque/force factor and $t_{m,i}$ is the dominant time constant of the system.

Introducing Eq. IV.1 in Eq. IV.4 or Eq. IV.5 would result in a model of the fifth order. However, the modern electric drives used in robotics have a typical cutoff frequency between 200 and 1000 Hz, which usually exceeds the first natural frequency of the manipulator. As long as the robot's closed-loop bandwidth remains at least one order of magnitude lower, the proportional approximation $\tau = k_t u$ is valid since the inner and outer loops build a *cascaded control* scheme.

However, this might not be the case for joints with low inertias as in wrists. In these cases, actuator dynamics dominate elasticity phenomena and Eq. IV.6 should be included in the rigid mode from Eq. IV.1, leading to a third-order model. Tam & al. [Tam91a] *examined solutions to this situation, which is not* considered further in this work.

IV.2.2 Decoupled Linear P(I)D Controllers.

A linear proportional-derivative (PD) controller can be designed to achieve exponential stabilization of a joint described either by Eq. IV.1, Eq. IV.4 or Eq. IV.5 (see Fig. 12).

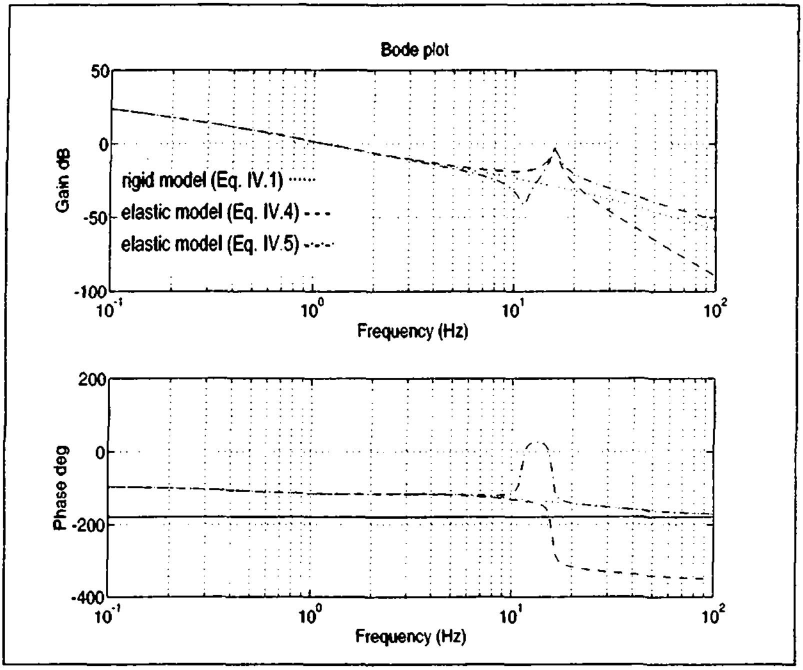


Fig.13 : Typical Open-Loop Frequency Responses of a Serial Arm Joint with PD control.

With the rigid model (Eq. IV.1), no apparent limit to the closed-loop bandwidth (and the corresponding gain) is posed. With the elastic model from Eq. IV.4, the gain must be low enough, so that the resonance is not excited. Things seem to change when Eq. IV.5 is considered since the closed-loop bandwidth may again exceed the natural frequency associated to elasticity. However, it should be stressed that the time response still contains a badly damped mode that may cause very large oscillations of $q_{l,i}$, even when $q_{m,i}$ is stable ! This is also discussed in § IV.5.5 in the case of robust control strategies.

In practice, it is therefore advised to place the sensors on the actuator's side of the elasticity and to close the loop on $q = q_m$ [Spon89].

Integrator versus Model-Based Gravity Compensation.

An integrator can be included in the control law to reject low-frequency perturbations such as gravity effects to obtain zero steady-state error at a very low cost. In practice however, this possibility is very limited because of robustness considerations (see Fig. 12).

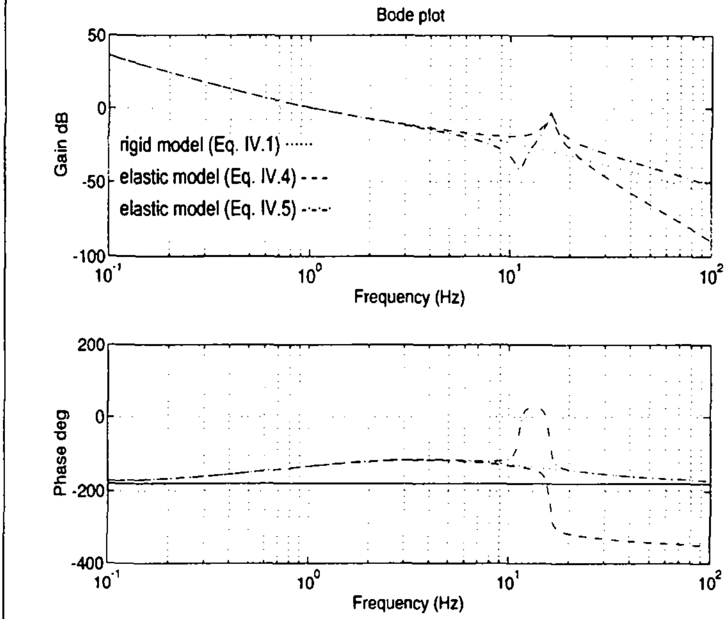


Fig.14 : Typical Open-Loop Frequency Responses of a Serial Arm Joint with PID control.

The presence of an integrator in the control law therefore poses both practical and theoretical problems:

- When *stick-slip* friction occurs, *limit cycles* appear.
- *Overshoots* are unavoidable if reference signals contain high frequencies.
- Since a PID is *not passive*, no formal stability proof can be obtained when nonlinear dynamics are considered (see § IV.5.3).

It is therefore often preferred to use PD control laws with model-based compensators, especially for the gravity [Baya88], [Wen 88]:

$$u = K_p\tilde{q} + K_d\dot{\tilde{q}} + \hat{g}(q), \qquad \text{(IV.7)}$$

where $\tilde{q} = q_c - q$ is the joint position error, K_p and K_d are diagonal matrices of constant gains and $\hat{g}$ an estimate of the torque/force vector due to the gravity.

Hats on symbols indicate that estimated, modeled or identified dynamics are used, which may somewhat differ from the true parameters of the manipulator. For the sake of simplicity, hats will be considered as implicit on the observed or measured state variables $q, \dot{q}$.

IV.2.3 The Model-Based versus Robust Control tradeoff.

On one hand, Eq. IV.7 might be regarded as a very simple model-based control law since a part of the inverse-dynamics model from Eq. III.15 should be computed in real time. On the other hand, it might be considered as a robust control law since the stability margins of the constant PD controllers are supposed to cope with the model uncertainties and perturbations caused by the neglected nonlinear dynamics.

In this respect, a slightly modified version of Eq. IV.7 might be considered, where the gravity compensation vector is a true feedforward command:

$$u = K_p\tilde{q} + K_d\dot{\tilde{q}} + \hat{g}(q_c)\,. \tag{IV.8}$$

This control law is very popular in today's industrial robots since $\hat{g}$ might be computed off-line, or *even obtained with a great accuracy for repetitive tasks through learning*[Tso 93]. However, it is intuitively clear (and it is formally proved in [Wen 88]) that the gains of the PD controller from Eq. IV.8 should be higher than those in Eq. IV.7 to cope with the perturbations introduced by the inaccurate compensation.

This is a simple illustration of the underlying tradeoff that can be found in the robot control literature: the computational burden of the controller can be reduced at the expense of high feedback gains that might excite unmodeled dynamics such as flexibility, while compensations of modeled nonlinear effects can reduce these gains but require a large computing power [Wen 88]. How to choose a good compromise is discussed in § IV.5.4.

Towards Really Nonlinear Robots.

The numerous disadvantages of reductions have led designers to consider lower ratios or even direct-drive transmissions, especially for small, light manipulators. In such robots, the nonlinear dynamics can no longer be neglected and the validity of the linear controller design above disappears.

However, various approaches from the nonlinear control theory presented below lead to control laws that are similar to the PD+gravity control from Eq. IV.7. In Section IV.3 and Section IV.5 below, relationships between existing nonlinear approaches and fundamental concepts from the linear control theory receive special attention.

IV.3 Model-Based Control.

IV.3.1 Feedback Linearization.

Pure model-based controllers rely on the use of the inverse dynamics model from Eq. III.15 to generate the command that would produce the desired move of the robot. The open-loop control law

$$u = {}^{q}\hat{A}(q) \cdot \ddot{q}_c + {}^{q}\hat{C}(q, \dot{q}) \cdot \dot{q} + {}^{q}\hat{g}(q) \qquad \text{(IV.9)}$$

is considered first, where $\ddot{q}_c$ is supposed to be available from the reference signals generator (see § II.10.3).

Combining Eq. III.15 and Eq. IV.9 in series results in the open-loop control scheme depicted in Fig. 15.

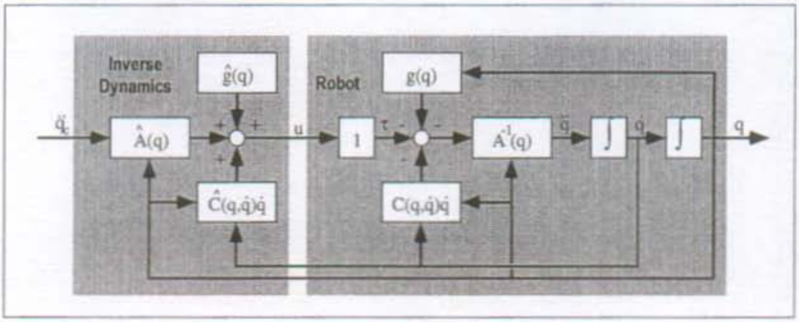


Fig.15 : Feedback Linearization in Joint Space.

In the ideal case where

$$\hat{A} = A \text{ , } \hat{C} = C \text{ and } \hat{g} = g \text{ ,} \qquad \text{(IV.10)}$$

the input-output relation becomes

$$q(t) = \iint \ddot{q}_c(t)\, dt. \qquad \text{(IV.11)}$$

In other words, the robot is seen through its inverse dynamic model as *linearized* and *decoupled* since each component of the output vector only depends on the corresponding element of the input vector.

From a more theoretical point of view, this approach is an application of the *feedback linearization* method, which is well known in nonlinear dynamics [Isid83]. The word "feedback" does not refer here to any control or stabilizing dynamics but simply to the fact that the system's state is used in the linearizing block, as indicated by the grayed arrows in Fig. 15.

IV.3.2 Closing the Loop : the Computed-Torque Scheme.

Markiewick [Mark74] proposed to exploit feedback linearization for the real-time control of robots. He called the corresponding scheme "*computed torque*" since the inverse dynamics model is used in the controller to compute the forces/torques that should be applied to the robot.

The heavy computational burden involved in the computed-torque approach as well as the limited performances of the microprocessors in the seventies prevented the development of this idea until 1980, when Luh, Walker & Paul developed the first efficient algorithm to compute the inverse dynamics model of any serial arm (see § III.7.2) [Luh80b]. Each of the n double integrators from Eq. IV.11 is controlled by an independent PD controller.

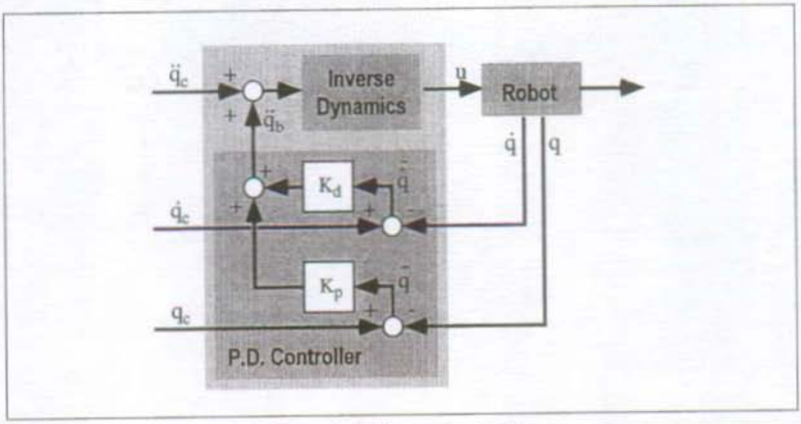


Fig.16 : Computed-Torque Control Scheme.

The whole scheme is represented in Fig. 15. The complete control law is given by

$$\begin{cases} u = \hat{A}(q) \cdot [\ddot{q}_c + \ddot{q}_b] + \hat{C}(q, \dot{q}) \cdot q + \hat{g}(q) \\ \ddot{q}_b = K_p \bar{q} + K_d \dot{\bar{q}} \end{cases} \qquad \text{(IV.12)}$$

This control law may be considered as a generalization of the PD+gravity control from Eq. IV.7 with three major extensions:

- the quadratic forces are also compensated;
- the joint acceleration reference signal is introduced in the feedforward command;
- the feedback gains are no more constant since they are multiplied by the inertia matrix $\hat{A}$, which depends on the manipulator's position.

IV.3.3 Computation of the Inverse Dynamics.

Adding inverse dynamics in the control law allows to keep the feedback gains low enough without limiting the tracking performance. The main problem in model-based control is the computational burden involved by the evaluation of the inverse dynamics model. Moreover, since its primary use is to generate *a feedforward command, this computation should be available at sampling intervals that correspond to the* reference signal spectrum. Since the reference signals are generated from a nonlinear process (see Section II.10), their spectrum may exceed the closed-loop bandwidth. This situation frequently arises in fast manipulators where the quadratic forces may become very important during short intervals (see the example of the Delta in Section V.6).

As a consequence, the inverse dynamics model must be computed at a higher frequency than twice the closed-loop bandwidth, which is the limit for the feedback control law according to Shannon's theorem. Some authors used slower sampling rates for the model computation than for feedback, arguing that the model does not vary faster than the robot's state. They are right only when quadratic forces are really negligible and when reference signals do not contain high frequency components. This is not the case for fast manipulators. (By the way, note that most simulation papers use sine reference signals!)

It is therefore an important issue in robot control to develop fast algorithms for the computation of the inverse dynamics model.

Reducing the Computational Burden.

An advantage of the Lagrange formulation over Newton-Euler's is that the model may be simplified by dropping negligible terms. Since most industrial manipulators move very slowly, quadratic forces are usually not computed in the model [Khos89b], [Tarn93]. In the same context, Aubin proposed to evaluate the energy of each of the computed torques/forces and to neglect the weakest terms [Aubi91].

Another approach is to compute the model off-line, based on reference values instead of the measured state. Bayard and Wen formally proved the validity of some schemes based on this idea [Wen 88]. They showed that higher feedback gains are required to stabilize the robot when reference values are used in place of measurements. Moreover, they mention that the Newton-Euler algorithm can either be computed with reference values only or with *measured state only, not a mix of both. As an efficient* alternative to the computed-torque scheme, the following control law was proposed:

$$u = \hat{A}(q_c) \cdot \ddot{q}_c + \hat{C}(q_c, \dot{q}_c) \cdot \dot{q}_c + \hat{g}(q_c) + K_p \tilde{q} + K_d \dot{\tilde{q}}, \tag{IV.13}$$

where the feedforward command can entirely be computed off-line while the feedback law is simply PD.

Note that such schemes require the task to be known enough in advance for to compute the inverse dynamics model. While this might be the case for repetitive tasks, modern applications often demand *instantaneous reaction and on-line generated* tasks. For this reason, it is unlikely that such control laws will be implemented on high-performance manipulators.

IV.3.4 Model-Based Control in Operational Space.

As mentioned in the introduction to this chapter, all the control laws presented in the survey above have an equivalent form in operational space. *It has been shown that the convergence properties of a* given scheme are the same in joint space and in operational space [Khat87], as long as the singular configurations of the manipulator are avoided.

In particular, model-based control laws in operational space can be derived from the feedback linearization of the end-effector's motions along the operational coordinates. In this case, the inverse *dynamics model from* Eq. III.43 is used. The corresponding scheme is given in Fig. 15 below. Note that the conversion of the robot's state from joint space to task space is performed by an observer, which is trivial for serial arms since $f(q)$ has a closed form. For parallel robots numerical techniques can be applied (*see § II.8.1*). *A direct measurement of the end-effector's position* would be highly desirable to solve this problem.

On the basis of this idea, Luh, Walker and Paul [Luh80a] developed the resolved-acceleration scheme, which is the equivalent of the computed-torque scheme in operational space. The corresponding control law is

$$\begin{cases} u = \hat{M}(p) \cdot [\ddot{p}_c + \ddot{p}_b] + \hat{D}(p, \dot{p}) \cdot \dot{p} + \hat{h}(p) \\ \ddot{p}_b = K_p \tilde{p} + K_d \dot{\tilde{p}} \end{cases}, \tag{IV.14}$$

where $\ddot{p}_c$ is the desired acceleration of the end-effector in operational space and p, $\dot{p}$ the state of the robot in operational space.

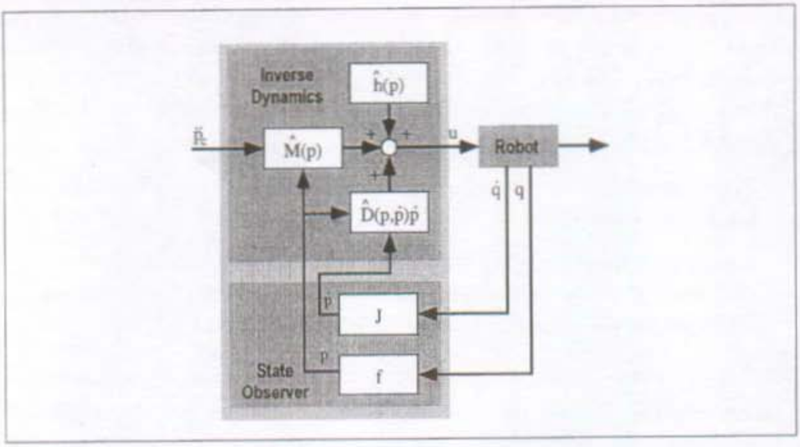


Fig.17 : Feedback Linearization in Operational Space.

Similarly, the equivalent to the PD+feedforward control law from Eq. IV.13 is

$$u = \hat{M}(p_c) \cdot \ddot{p}_c + \hat{D}(p_c, \dot{p}_c) \cdot \dot{p}_c + \hat{h}(p_c) + J^T(p_c)\left[K_p\tilde{p} + K_d\dot{\tilde{p}}\right], \qquad \text{(IV.15)}$$

where $J^T(p_c)$ is the Jacobian matrix computed from the reference position.

IV.3.5 Advantages and Drawbacks.

A control law in operational space can be considered as a reformulation of the equivalent law in joint space through a nonlinear change of coordinates. One should therefore not expect to improve radically the robot's performance thanks to an operational space scheme. However, control in operational space has two major advantages:

- Since the desired task is defined in operational space, a scheme in operational space controls the variables of interest to the user. Effects of the nonlinear kinematics of the manipulator are cancelled and, when the nonlinear dynamics are also compensated, constant dynamic behavior of the end-effector can be expected over the whole workspace.
- Such a constant behavior is highly desirable in applications where the control of the force exerted by the end-effector on the environment is required. In this case, the forces/torques applied on the end-effector can easily be included in the inverse dynamics model in operational space (Eq. III.43), while the joint space model (Eq. III.15) is less suitable.

From the implementation point of view, the computation burden associated to the evaluation of the inverse dynamics model of a serial arm seems to be higher in operational space than in joint space when only Eq. III.15 and Eq. III.17 are considered.

However, a fair comparison should consider also that joint space schemes require a conversion of the reference signals (see § II.10.3). While this might be computed off-line when repetitive tasks are to be performed, modern applications often require the references to be generated on-line. In this context, *Dombre & Khalil [Domb88] showed that the resolved-acceleration control law* (Eq. IV.12) *is in fact less* expensive than the computed-torque control (Eq. IV.12) for serial arms because:

- the evaluation of the forward kinematics relation f is required instead of f^{-1}, which is more difficult to compute for serial arms (see § II.8.2);
- in the computed-torque scheme, the Jacobian has to be computed from the reference values while the inertia matrix is computed from the robot's measured position, preventing to take advantage of the redundancy between kinematics and dynamics computations (see § III.8.4).

When the PD+feedforward scheme is considered, the second advantage disappears. Moreover, the amount of real-time computations required in the joint space scheme (Eq. IV.13) is considerably increased by the state observer and the multiplication by J^T required in the operational space scheme (Eq. IV.13).

IV.3.6 Reported Experimental Results.

Table 6 below summarizes some reported experimental results obtained with model-based control strategies on serial arms.

All these contributions reported considerable improvement with model-based control over independent PD control that was systematically used for comparison. Interestingly, no significant *difference could be found between the performance of the PD + feedforward scheme and the computed-*torque scheme.

Tam & al. obtained also comparable performance with the *resolved-acceleration scheme in* operational space. They used a simplified inverse dynamics model where quadratic forces were neglected and the inertia matrix considered as diagonal and argued that the higher sampling rate allowed by these time-saving approximations compensate for the loss of accuracy in the model. However, the experiments of Leahy & al on the same type of robot showed the importance of implementing the compensation of the quadratic forces even at low velocities.

Author	Robot	Control Unit	Control Law
An& al. [An89]	SLDDA (MIT 3-DOF direct-drive)	68000 h=7.5 [ms]	computed-torque Eq. IV.12
"	"	68000 h=7.5 [ms]	joint PD + feed-forward (ff) Eq. IV.13
Khosla & Kanade [Khos89b]	CMU DD Arm II (3-DOF direct-drive SCARA)	68000 + 3 x TMS320 h=2 [ms]	computed-torque Eq. IV.12
Leahy & al. [Leah89]	PUMA 560 (industrial 6-DOF spherical)	?	computed-torque Eq. IV.12
Tam & al [Tam91b], [Tam93].	PUMA 560 (industrial 6-DOF spherical)	Microvax II + 6 x special hard. h=5.41 [ms]	joint PD + ff Eq. IV.13
"	"	"	computed-torque Eq. IV.12, $\hat{C} = 0$
"	"	"	resolved-acceleration Eq. IV.14, $\hat{D} = 0$

Table 5 : Reported Experimental Setups for Model-Based Control of Serial Robots.

IV.4 Adaptive Control.

The tutorial paper from Ortega and Spong [Orte89] is considered as the best review to date on adaptive approaches to robot control. These authors consider as adaptive the control laws where parameter estimation is explicitly included. A distinction is made between adaptive inverse dynamics schemes and passivity preservation schemes.

IV.4.1 Adaptive Inverse Dynamics.

In this approach, the inverse dynamics model used in a computed-torque based control is identified in real time. The first such scheme has been proposed by Craig, Hsu and Sastry in 1986 [Crai86]. It assumes that the functional form of the estimated inertia matrix $\hat{A}$, quadratic forces matrix $\hat{C}$ and gravity vector $\hat{g}$ is known and that only the parameters φ from Eq. III.14 have to be estimated. It is then showed that the error dynamics is given by

$$K_p\tilde{q} + K_d\dot{\tilde{q}} + \ddot{\tilde{q}} = \hat{A}^{-1} \cdot Y \cdot [\hat{\varphi} - \varphi], \qquad \text{(IV.16)}$$

and therefore that an update law obtained from a Lyapunov approach can be found that guarantees global convergence of the whole scheme.

However, measurements of the joint acceleration $\ddot{q}$ as well as the boundedness of the inverse of the estimated inertia matrix are required. This second drawback was removed in [Spon90], where the computed torque is implemented using fixed estimates of the inverse dynamics model like in Eq. IV.12, but an additional term that depends on the identified parameters $\hat{\varphi}$ is added to $\ddot{q}_b$.

Finally, the requirement of measured joint acceleration has been removed by Middleton and Goodwin [Midd88] through very clever use of filtering combined with nonlinear dynamics.

IV.4.2 Passivity-Based Control Methods.

In these approaches also, the property of the manipulator's dynamics to be linear in its parameters is exploited and the vector of identified parameters $\hat{\varphi}$ is obtained in real time through a gradient-type updating algorithm. However, using the control law from Eq. IV.18, the resulting adaptive scheme does not require accelerations measurements or additional hypothesis on the identified parameters [Orte89].

The different schemes that follow this approach mainly differ by the type of the $K(s)$ transfer function, which is used in the control. The most well known is the one from Slotine and Li [Slot87a] where $K(s) = sK_i$, with K_p being a diagonal matrix of positive gains. The same authors also proposed a way to speed up the slow convergence rate obtained with gradient update techniques as in Craig's approach [Crai86]. Least-squares estimates are known to be locally faster, but they unfortunately lack the required passivity property. For this reason, Slotine and Li developed a modified least squares algorithm that uses both the tracking error and the prediction error [Slot87a].

IV.4.3 Simulation Results and Open Problems.

Schwartz [Schw90] compared the adaptive schemes from Craig [Crai86] and from Slotine and Li [Slot87a] for persistently and non-persistently exciting trajectories, with and without measurement noise on a simulated 2-DOF serial arm. The main results are that Craig's method is slow (thousands of samples before convergence) and inapplicable when measurements are noisy. Slotine and Li algorithm was shown to suffer from the same problems, although convergence was a bit faster (hundreds of samples before convergence) for noiseless measurements.

These serious limitations to the practical use of adaptive control laws arise from some theoretical open problems. First, the transient performance is not addressed in the proposed approaches, nor do they prove that the asymptotic convergence of the closed-loop system is uniform. It is therefore possible that small changes in the dynamics may result in loss of stability, which can also happen in the presence of bounded disturbances [Reed89]. This sensitivity to disturbances can be overcome by requiring persistently exciting signals to achieve uniform convergence of the parameters. However, this is not a realistic solution from the practical point of view.

IV.4.4 Experimental Results and Conclusion.

While many theoretical contributions to the adaptive control of manipulators are published every year, only a handful of actual experimental results is available [Crai88], [Niem88], [Slot87b]. The obtained convergence rates are so slow that application of the corresponding methods to fast manipulators is out of question.

Much better results were obtained with the indirect adaptive method in which the parameters of a computed torque controller are identified with a standard forgetting factor least-square algorithm. This approach is not validated by any convergence proof but converged even in the presence of noise [Schw90] and gives the best results in practice! Fässler obtained high convergence rates with this approach by considering only one varying parameter and taking advantage of the particular nature of the manipulator (a 3-DOF cylindrical arm) [Faes88].

Adaptive schemes require very high quality measurements and large computing power that are not acceptable from an economical point of view as long as the performances are not shown to stand higher than those obtained through model-based and robust control. Adaptive approaches are therefore not considered further in this work.

IV.5 Robust Control.

Abdallah & al. gave a very comprehensive survey on robust control in robotics in which three different approaches are distinguished : linear multivariable, variable structure control and passivity [Abda91].

IV.5.1 The Linear Multivariable Approach.

The linear multivariable approach allows to design a linear controller that replaces the n decoupled PD in the computed-torque control so that is satisfies specified performance criteria in presence of parameter variations. *Bounds on the inertia matrix and on the model mismatch should be available to the* designer, but too conservative bounds lead to high gain feedback.

Becker & Grimm showed that controllers based on H_∞ theory can easily be obtained to obtain bounded L_∞ norm of the tracking errors [Beck88]. However, similar results for L_2 theory require reformulation of the problem and additional assumptions such as noiseless measurements.

Freund used a pole-placement approach that ensures stability when the robot's dynamics stay within the given bounds [Freu82]. Fundamentally, the resulting control law is the same as Eq. IV.12 with a polynomial structure instead of the PD.

IV.5.2 Variable Structure Controllers.

The theory of *variable structure* (VSS) has been applied to the control of many nonlinear processes, including robots since the work of Young in 1978 [Youn78]. *The main feature of this approach is to drive* the error to a *switching surface* on which the system is in *sliding mode*, unaffected by modeling uncertainties and disturbances.

Unfortunately, for the schemes based on pure VSS theory, the control law is discontinuous with respect to the error signal and therefore creates *chattering* which excites high-frequency dynamics such as elastic modes. To address this problem, Slotine proposed the "stuction control" [Slot83], [Slot85] where the controller is smoothed inside a *possibly time-varying boundary layer. In this case, the control law is* given by

$$u = \hat{A}(q) \cdot \left[\ddot{q}_c + K^2\tilde{q} + 2K\dot{\tilde{q}} + \phi(q, \dot{q}, t)\right] + \hat{C}(q, \dot{q}) \cdot \dot{q} + \hat{g}(q), \qquad \text{(IV.17)}$$

where K *is a diagonal matrix of constant gains and* ϕ a nonlinear term determined by the extent of the parametric uncertainties [Slot85].

This control law exhibits the same structure as the computed-torque controller Eq. IV.12 but introduces robustness in a more formal way. However, asymptotic stability of the error can not be shown as in passivity based approaches.

IV.5.3 The Passivity Approach.

The *passivity theorem* [Deso75] guarantees that a closed-loop system is asymptotically stable if each of the input-output mapping of each block in the loop is passive. Thus, since the $\tau \to \dot{q}$ mapping is passive for any manipulator (see § III.5.1 and § III.8.2), any controller with a passive mapping $\dot{q} \to \tau$ asymptotically stabilizes the closed-loop system.

However, this result guarantees that $\dot{\tilde{q}} \to 0$ as $t \to \infty$, but not that $\tilde{q} \to 0$. Two possible solutions are detailed below.

Using a Filtered Version of the Error Signal.

A passive map $\tau \to \rho$ exists for any robot, where $\rho = [sI + (K(s))/s] \cdot \tilde{q}$ is a filtered version of the error signal $\tilde{q}$ [Orte89].

When $K(s)$ is chosen such that $[sI + K(s)/s]^{-1}$ is a strictly proper, stable transfer function, it may be shown [Orte89] that both $\dot{\tilde{q}}$ and $\tilde{q}$ are asymptotically stable with the control law

$$u = \hat{A}(q) \cdot [\ddot{q}_c + K(s)\tilde{q}] + \hat{C}(q, \dot{q}) \cdot \left[\dot{q}_c + \frac{K(s)}{s}\tilde{q}\right] + \hat{g}(q) + K_v \cdot \left[sI + \frac{K(s)}{s}\right]\tilde{q}, \quad \text{(IV.18)}$$

where K_v is a positive definite constant gain matrix.

This approach is widely used in the design of adaptive controllers based on the passivity property, which will be discussed in § IV.4.2. However, its use for the design of fixed robust controllers is not obvious.

Using LaSalle's Theorem for Regulation.

Another approach is to consider the control law

$$u = u_f + K(s)\dot{\tilde{q}}, \quad \text{(IV.19)}$$

where $K(s)$ *is strictly positive real (SPR) and* u_f *is* L_2 *bounded.*

Note that the inclusion of an integrator in $K(s)$ to reconstruct $\tilde{q}$ destroys the SPR condition. By an appropriate choice of $K(s)$ and u_f, the passivity theorem can be applied to show that $\dot{\tilde{q}}$ and u_b are bounded in the L_2 norm. Since $K^{-1}(s)$ is SPR, $\dot{\tilde{q}}$ is asymptotically stable.

However, the development above implies that the error $\tilde{q}$ is bounded, but not asymptotically stable in the case of time-varying references. Using LaSalle's theorem [Spon89], it is nevertheless possible to obtain asymptotic stability of $\tilde{q}$ when the gravity is perfectly compensated through a suitable feedforward control. The corresponding control law is then

$$u = g(q) + K(s)\dot{\tilde{q}}. \qquad \text{(IV.20)}$$

This approach is said robust since stability is guaranteed regardless of the robot's parameters, as long as $K(s)$ is strictly positive real.

In this context, Anderson [Ande89] showed that an *a priori knowledge of the variations of the inertia* matrix can be exploited in a control law of the form

$$u = K_p(q)\tilde{q} + K_d(q)\dot{\tilde{q}} + \hat{g}(q_c), \qquad \text{(IV.21)}$$

where the diagonal gain matrices $K_p(q)$ and $K_d(q)$ are obtained from the singular values of the modeled inertia matrix $\hat{A}(q)$.

The same kind of reasoning led to the proof of the globally asymptotically stability of the linear PD control on each joint, provided that the gravity effects are compensated, as in Eq. IV.7 [Pade88].

IV.5.4 A Unified Point of View Thanks to Lyapunov.

Bayard and Wen established a convenient lemma to handle third-order terms in the Lyapunov function derivative for robot manipulators [Wen 88]. This avoids the need for a generalization of the invariance principle to time-varying systems and therefore replaces LaSalle's theorem when addressing varying reference signals. Bayard and Wen could prove the global exponential stability of a dozen of non-adaptive control laws for the robot tracking problem [Wen 88] and obtain the corresponding lower bounds of the feedback gains. Similar results were obtained for seven adaptive laws using the very same theory [Baya88].

Many of the control laws mentioned above belong to the set examined by Bayard & Wen, who clearly showed that each one provides a compromise solution to the tradeoff between computation and convergence properties.

IV.5.5 Limitations of High Gain Feedback Control Laws.

Purely robust control laws that do not use any inverse dynamics model evaluation are most often proposed by researchers with an electrical engineer background. Since the power amplifiers of DC motors use switching transistors to control the currents in the motor's coils as a function of its position, it seems reasonable to modify the control law to achieve acceleration, speed or even position control that would be robust with respect to inertia variations and external perturbations.

Obtaining a high quality estimation of the motor's speed is the main problem in practice [Canu91], which requires the most part of the computing power of the digital signal processor (DSP) traditionally used for these applications. Moreover, it should be stressed that no transmission delays are tolerated in such schemes, requiring the DSP to work at a very high sampling rate and directly connected to the power amplifier. In this case, the closed-loop bandwidth is theoretically only limited by the actuator's dynamics (see § IV.2.1), reaching hundreds of Hertz.

However this approach is not realistic in practice if the elastic and flexible modes of the manipulator are not taken into account in the controller's synthesis. Even if these dynamics are not directly in the control loop, their reaction on the actuator can be high enough to cause an unstable behavior of the mechanical structure while the controlled loop keeps being stable [Eppi93]. Note that the elastic modes are often weakly damped because of the low frictions and, most important, that their natural frequency varies with the manipulator's inertia, preventing to use standard *notch filters* as for machine-tools.

For this reason it is claimed here that all the results published in the field of variable structure, high gain or robust control of robots should be examined vary carefully : applied high gain control schemes must explicitly consider at least the joint elastic mode and therefore run into the problems described in § IV.2.1. In all the available experimental results, the gains are reduced so that the controlled bandwidth is limited below these natural frequencies and they therefore belong to the classical control schemes mentioned in Section IV.2. In this case, there is no need of sampling periods shorter than a few milliseconds.

This rather polemical assertion is somewhat supported by the fact that no experimental result is available that show the effectiveness of a purely robust controller on an industrial manipulator. In this context, it can also be mentioned that some important robot manufacturers produce their own actuators, power amplifiers and controllers, but do not use such techniques.

IV.6 A Control Strategy for Fast Parallel Manipulators.

IV.6.1 Applications of Parallel Manipulators and Associated Control Schemes.

As mentioned in § I.5.2, many parallel structures are used as strong, rigid mobile platforms. Their stiffness has then been exploited in tasks requiring force control while the Delta robot initiated their use when speed is the major consideration. Three distinct approaches to the control problem appeared successively:

- In flight-simulator cabins, the goal is to create artificial accelerations of the platform. The equivalent accelerations of the joints are obtained through Eq. II.31. Additional slow dynamics are added to re-center the platform in its workspace. Since the platform's inertia varies little, a joint space control scheme based on a PD+gravity control law gives satisfactory results [Stew65]. Since only inverse kinematics is required, parallel structures have an advantage over serial arms when the computational burden is considered.
- When a certain force has to be exerted by the end-effector on the environment, the mapping onto joint forces is made through Eq. III.2, where the Jacobian is involved. On parallel manipulators, the inversion of the inverse Jacobian is required (see § II.9.6). Moreover, since robots in contact with their environment move very slowly, the effects connected to their dynamics are negligible and a static model is sufficient in practice. The advantages of a control scheme in operational space enumerated in § IV.3.5 become evident in this context and such approaches gave excellent practical results [Merl88].
- As for serial arms, achieving high performance for faster manipulators with low reduction rations require full compensation of their dynamics. The Delta even initiated a series of parallel manipulators that are so fast that the magnitude of the quadratic forces can exceed all other dynamics (see Chapter V). The very few available contributions to the control of such manipulators are detailed in § IV.6.3.

IV.6.2 Simplified Models for Fast Manipulators.

In most practical cases, the link's inertia is negligible with respect to the other bodies. In this case, the inertia matrices ${}^{q}A$ and ${}^{p}A$ become constant, block diagonal or even diagonal. The quadratic forces are therefore zero and the inverse dynamics model reduces to

$$\tau = {}^{q}A \cdot \ddot{q} + {}^{q}g(q) + J^{T}\left[{}^{p}A \cdot \ddot{p} + {}^{p}g(p)\right]. \qquad \text{(IV.22)}$$

Such simplified models have been developed and used for the Delta robot [Codo91a], [Codo91b], the Hexa [Pier91], [Pier92], [Pier91], and Speed-R-Man [Rebo90], [Rebo91].

IV.6.3 Reported Experimental Results.

Because of the lack of systematic approach to model parallel manipulators, it is difficult to compare objectively the few experimental results obtained on the various manipulators. The results listed in Table 6 use model-based control laws that depend on the considered robot.

Author	Robot	Control Unit	Control Law
Codourey [Codo91a], [Codo91b]	Delta (3-DOF fully parallel, direct-drive)	3xT800 Transputer h=7.5 [ms]	joint PD + ff Eq. IV.13
Kokkinis & al. [Kokk91a], [Kokk91b]	CRSM 5YR (3-DOF spatial 5 bars, direct drive)	80286 + AT&T DSP32 h=1.7 [ms]	joint PD + ff
“	“	“	operational PD + ff
“	“	“	resolved-acceleration (see comment below)
Karidis, Zai & al [Kari92], [Zai92]	IBM “Hummingbird” (3-DOF plane 5 bars, direct drive)	3 x T800 Transputer h=0.125 [ms]	computed-torque Eq. IV.12, $\hat{C} = 0$

Table 6 : Reported Experimental Setups for Model-Based Control of Parallel Robots.

While joint space PD control gives satisfactory performance on geared industrial manipulators, Codourey obtained noteworthy improvements by adding a feedforward command [Codo91a], [Codo91b].

IBM's Hummingbird micropositioner has a plane 5 bars structure that allows an inverse dynamics model in joint space only to be obtained. Its accelerations of about 50 G are of the same order of magnitude than those obtained with the Delta, but the speeds are much lower since the Hummingbird's workspace is only 13x13x1 [mm]. The quadratic forces are therefore neglected in the computed-torque control scheme that is computed at the astounding rate of 8 KHz [Kari92], [Zai92].

Kokkinis, Nakamura and Uecker developed a control scheme in operational space that is similar to the one proposed in this work, except that the free joints are explicitly used in the model (see § III.6.1) [Kokk91a]. They obtained about the same performance as with the joint PD + feedforward control law.

A Note about Robust and Adaptive Approaches.

Some results with robust control laws on parallel robots are reported. These strategies meet the same problems as for serial arms (see § IV.5.5) : since no filtering of the natural frequencies of the structure is done, it is suspected that the behavior of the whole structure is not as good as the nice performance displayed in joint space.

To the best of the knowledge of the author, the only adaptive law specifically developed for parallel manipulators by Walker [Walk90] has not been experimentally tested.

IV.6.4 Joint Space Versus Task Space Control Schemes.

The computed-torque scheme and resolved acceleration scheme for parallel robots are displayed respectively in Fig. 18 and Fig. 19.Both schemes require to:

- evaluate the forward kinematics relation (Eq. II.22),
- compute the inverse Jacobian matrix (Eq. II.16),
- invert it to get the Jacobian (Eq. II.19),
- evaluate the forward speed kinematics relation (Eq. II.15),
- compute the inverse Jacobian's derivative (Eq. II.31),
- apply the PD. (or robust) control law,
- *calculate the acceleration mapper,*
- apply the PD (or robust) control law,
- *evaluate the inverse dynamics model in the two spaces (see § III.6.2 and § III.7.4).*

Therefore, as displayed in Fig. 18 and Fig. 19, both schemes require exactly the same amount of computations. However, the computed-torque scheme requires conversion of the reference signals from task space to joint space, as discussed in § II.10.3. Therefore, the discussion in § IV.3.5 about the advantages of the task space control scheme for serial arms remains perfectly valid when parallel manipulators are considered.

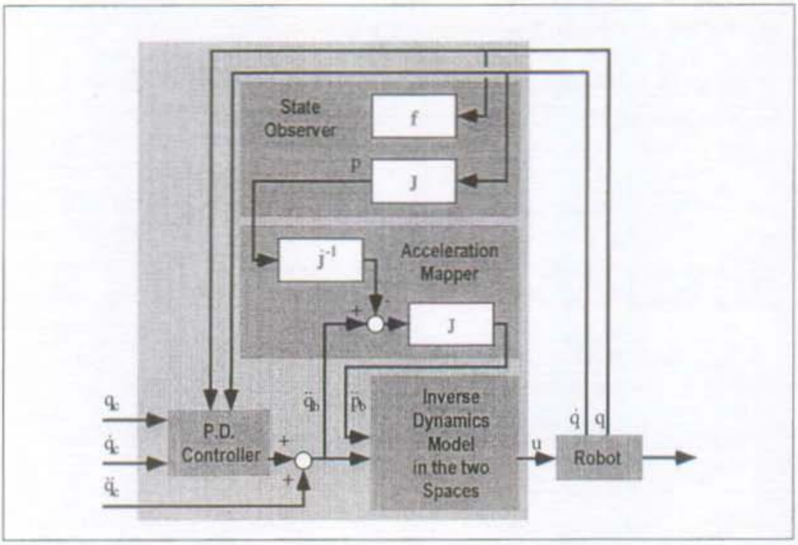


Fig.18 : Computed Torque Scheme for a Parallel Manipulator.

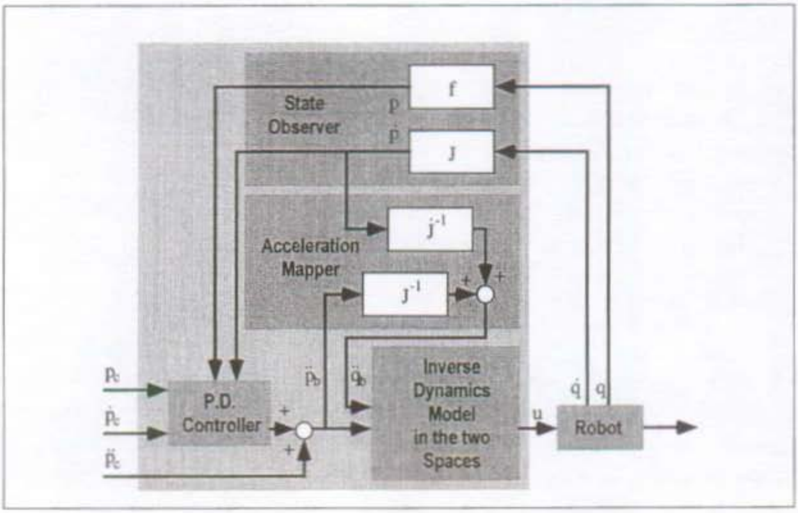


Fig.19 : Resolved-Acceleration Scheme for a Parallel Manipulator.

IV.7 Implementation Considerations.

IV.7.1 Previous Work on Controllers for Serial Arms.

How to design the hardware and software of an advanced robot control unit has been investigated by many researchers. Many results are available for serial arms controllers.

In the early 80's, Luh and Lin studied how to distribute optimally the computational load among processors working in parallel [Luh80b]. This is known to be an NP-complete problem [Bokh88], [Sark89] for which a solution can be found only when the computational load is divided in a small number of parts to be assigned to processors. Such an *coarse grain* parallelism is known to lead to poorly efficient solutions, where *efficiency* is defined as

$$\text{efficiency} = \frac{\text{execution time on a single processor}}{\text{execution time in parallel} \times n}. \qquad \text{(IV.23)}$$

Efficiency is a measure of the average business rate of the processors working in parallel.

A more promising approach is to exploit the knowledge of the *fine grain* structure of the algorithms either by using multiprocessor controllers or by designing specific electronic circuits. A (non exhaustive) list of important contributions to these two trends is given below. More references can be found in Graham's excellent survey on computer architecture for robotics [Grah89].

Parallel Software Design.

Parallel evaluation of inverse kinematics has been studied by Sugimoto [Sugi89], Tourassis [Tour89] and Zhang & Paul [Zhan91]. Wander and Tesar proposed a pipelined computation of the inverse dynamics [Wand87], while parallel implementation was investigated by Chen and Lee [Chen88], Fijany [Fija88], [Fija89] and Zomaya & Morris [Zoma90], [Zoma92], who implemented their algorithms on Transputers. The use of massively parallel systems based on systolic networks has been examined by Javaheri and Orin for the evaluation of the inertia matrix [Java88].

Hardware Design.

Special VLSI circuits were developed by Chang & Lee to compute the inverse kinematics relation [LeeG87] or the pseudo-inverse Jacobian for redundant arms [Chan89] and by Hsia & al to compute the Jacobian together with its derivative and inverse [Hsia91]. A restructurable VLSI vector processor based on three floating-point processing units (FPU) was proposed by Sadayappan & al. as a basis of a general robot control unit [Sada89]. Ish-Shalom and Kazanzides developed a control unit based on multiple digital signal processors (DSP) working in parallel [Ish-89].

A Comment on the use of DSP's

Interestingly, using a single DSP to control a robot is not a widespread idea, which is mainly supported by Kokkinis' work [Kokk91a], [Kokk91b]. The reason is probably that such processors are specialized devices that are optimized to perform simple, linear operations such as scalar products on data flowing "through" the processor at a very high rate.

In robotics, the bandwidth of the input and output signals does not exceed 1 KHz. Vector and matrix operations represent only a small fraction of the total computations, while several transcendental functions have to be evaluated.

A single DSP is therefore better fitted to control the inner loop in a power amplifier rather than the outer loop discussed in this work.

IV.7.2 Parallel Controllers for Parallel Robots.

As shown in Chapter II and Chapter III, the structure of the algorithms involved in the computation of kinematics and dynamics of parallel manipulators is very different from the serial case. As a consequence, the results mentioned in § IV.7.1 can not be used without deep modifications.

The parallelism of the proposed model-based control scheme in operational space is first analyzed from a *coarse grain* perspective, then its *fine grain* structure is exhibited and a matching *parallel+pipeline structure* of the control unit is proposed.

Coarse Grain Structure.

The different blocks of Fig. 19 have to be evaluated during each sampling interval. Their dependencies force this evaluation to follow a sequence that is depicted in Fig. 20. With this representation, the computation of each block can only take place when all the superimposed blocks have first been evaluated.

It can therefore be seen that the complete algorithm has a dominant sequential structure, where only the control law can possibly be computed at the same time as J^{-1}, while the generation of the reference signals can be done in parallel with the sequence of kinematics computations.

This coarse grain analysis shows that an implementation on two processors might be considered, *but that the efficiency of such a setup is likely to be poor because of the dissymmetry of the computational* burden assigned to each processor.

The possible sampling period is therefore limited by the *traversal time* of the data through the most loaded processor :

$$t_s = t_1 + t_2 + t_3 + t_4 + t_5 . \qquad \text{(IV.24)}$$

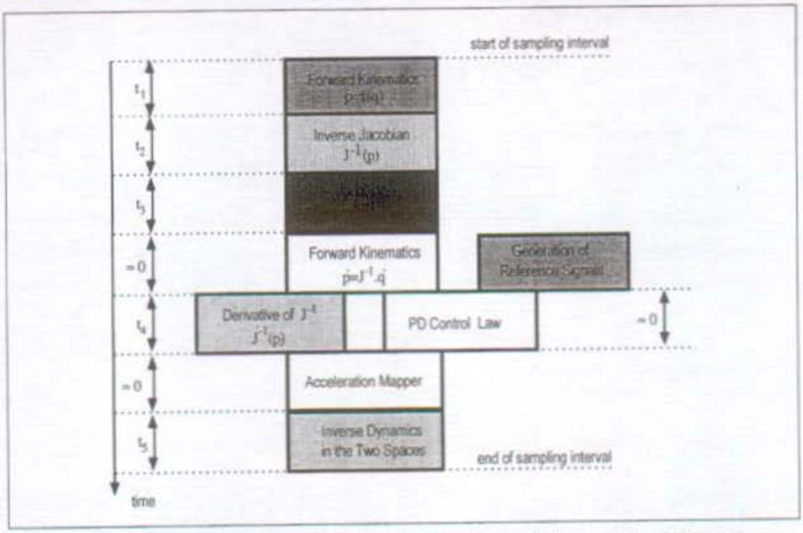


Fig.20 : Coarse Grain Structure of the Resolved Acceleration Scheme for Parallel Robots.

Fine Grain Structure.

Examining Fig. 20 closer shows that the blocks in light gray correspond to algorithms that have an intrinsic parallel structure over the n dimensions of either the joint space or operational space. In other words, these blocks might be divided in n independent blocks that can be computed in parallel.

White blocks are also parallelisable, but the corresponding operations are so simple that the computational time that could be saved is counterbalanced by the overhead caused by the communication of data between the processors. Their traversal time is even negligible with respect to the other blocks. It is therefore not worthwhile to parallelize these blocks.

Dark gray blocks are not parallelisable. In fact, the inversion of a matrix is, but the involved techniques are not worth to be used for small n as in robotics applications.

Medium gray blocks have a structure that combines some parallelisable parts with non-parallelisable parts. The generation of reference signals contains the profile generator, which is not parallelisable and a geometric part that can be parallelised according to the dimensions of operational space. The forward kinematics relation has a similar structure in the rare cases where it has a closed form (Delta). The more general iterative *Newton-Raphson*-based algorithm described in § II.9.6 has an inherent sequential+parallel structure. Exploiting this fine grain parallel structure leads to the situation shown in Fig. 21 below.

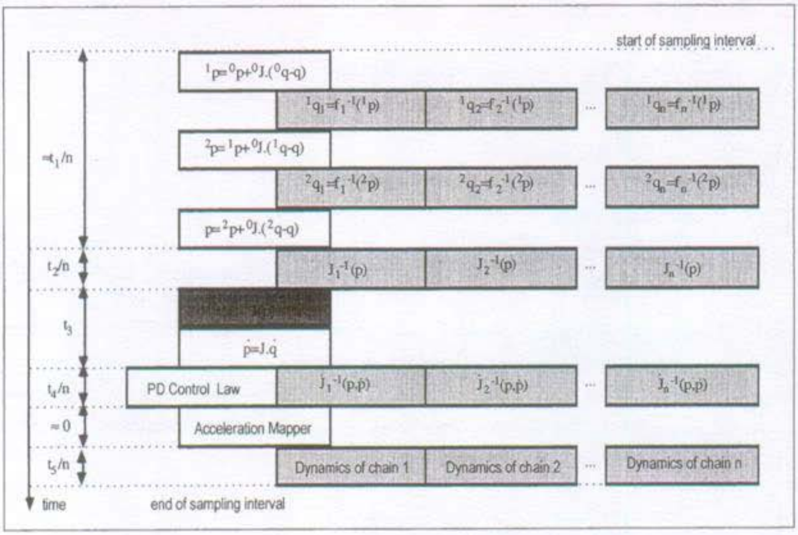

start of sampling interval
$^1p = {}^0p + {}^0J.({}^0q - q)$
$^1q_1 = f_1^{-1}({}^1p)$
$^1q_2 = f_2^{-1}({}^1p)$
...
$^1q_n = f_n^{-1}({}^1p)$
$^2p = {}^1p + {}^0J.({}^1q - q)$
$^2q_1 = f_1^{-1}({}^2p)$
$^2q_2 = f_2^{-1}({}^2p)$
$^2q_n = f_n^{-1}({}^2p)$
$\approx t_1/n$
$p = {}^2p + {}^0J.({}^2q - q)$
t_2/n
$J_1^{-1}(p)$
$J_2^{-1}(p)$
$J_n^{-1}(p)$
t_3
$\dot{p} = J.\dot{q}$
t_4/n
PD Control Law
$\dot{J}_1^{-1}(p,\dot{p})$
$\dot{J}_2^{-1}(p,\dot{p})$
$\dot{J}_n^{-1}(p,\dot{p})$
≈ 0
Acceleration Mapper
t_5/n
Dynamics of chain 1
Dynamics of chain 2
Dynamics of chain n
time
end of sampling interval

Fig.21 : Fine Grain Parallel Structure.

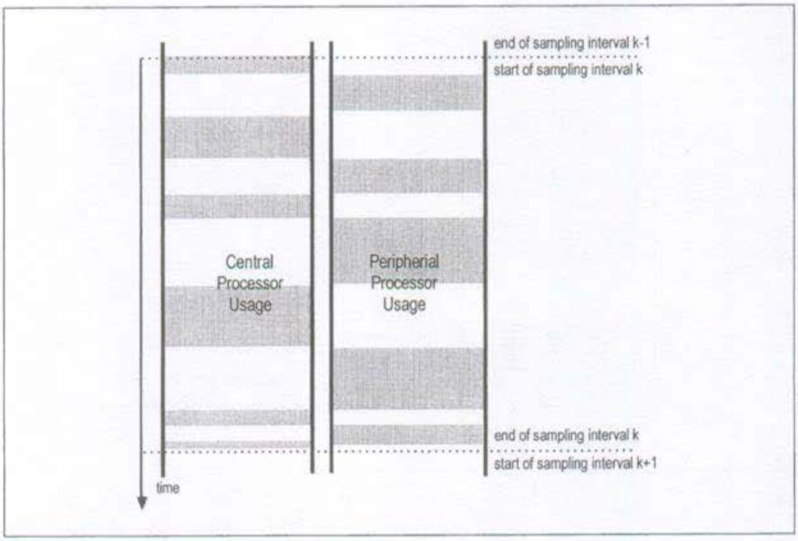

end of sampling interval k-1
start of sampling interval k
Central Processor Usage
Peripherial Processor Usage
end of sampling interval k
start of sampling interval k+1
time

Fig.22 : Processor Usage over a Single Sampling Interval.

The traversal time is then

$$t_p \equiv t_3 + \frac{t_1 + t_2 + t_4 + t_5}{n}. \qquad \text{(IV.25)}$$

Design of a Parallel + Pipeline Control Unit.

The main fact that appears in Fig. 21 is that the white and dark gray blocks can be implemented on a *central* processor while the light grayed blocks can be implemented on n *peripheral* processors that exchange data only with the central processor. The topology of the processors net is therefore a star.

Moreover the sequential structure found in the coarse grain analysis has the following interesting consequence: the central is busy while the peripherals are idle, and the central waits while peripherals compute. This is illustrated in Fig. 22, were the CPU usage of both types of processors is displayed over a single sampling interval. The sequence of operations corresponds to Fig. 21 where realistic proportions between the computation times were introduced.,

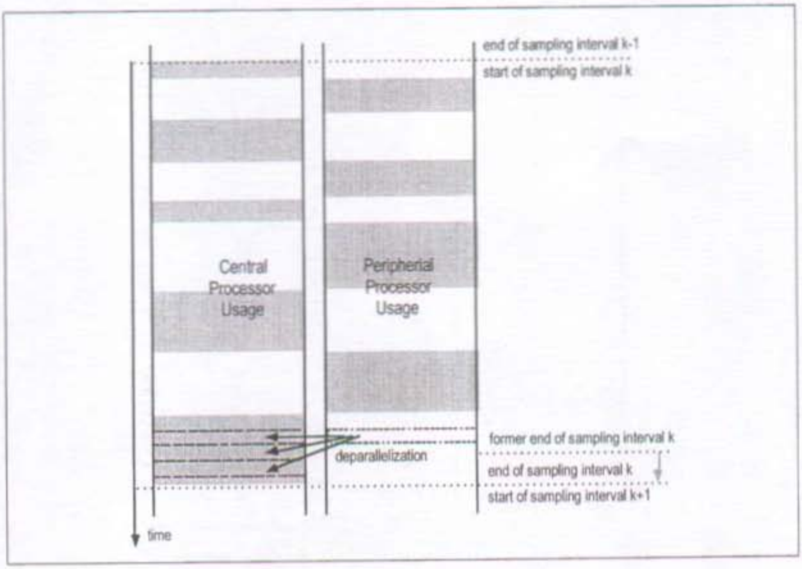

end of sampling interval k-1
start of sampling interval k
Central Processor Usage
Peripherial Processor Usage
former end of sampling interval k
deparallelization
end of sampling interval k
start of sampling interval k+1
time

Fig.23 : Obtaining a Balanced 50% CPU Usage.

Examining Eq. IV.25, one may remark that, if t_3 is close to $(t_1 + t_2 + t_4 + t_5)/n$, all the $n + 1$ processors are idle about half of the time.

If the central processor is too busy, meaning that the peripherals are under-utilized, the designer is suggested to take the decision of reducing the number of peripherals by grouping the tasks that were assigned to two of them on a single one. This is very likely to occur in controllers for 6-DOF fully parallel robots, where one peripherial processor would typically be associated to 2 DOFs.

If the central processor is less than 50% busy (as represented in Fig. 22), the designer can choose to de-parallelize an algorithm and move it from the peripherals to the central processor. This is illustrated in Fig. 23, which is obtained from Fig. 22 by moving a small process from the (three) peripherial processors to the central one. As in the previous case, the sampling period has to be lengthened by a factor that remains smaller than 2.

Once all the processors are busy about 50% of the time, a two-stage software pipeline structure can be introduced that allow to double the sampling frequency at the expense of adding one sampling period of delay in the feedback loop. The original idea proposed here is to interleave two consecutive sampling periods so that the second one begins while the first one is not yet terminated. The traversal time of the algorithm remains the same, but covers two sampling periods instead of one. Since the processors are now 100% busy this shows the perfect efficiency of the proposed parallel control unit, as shown in Fig. 24 where the computations associated to successive sampling intervals are represented by different gray levels.

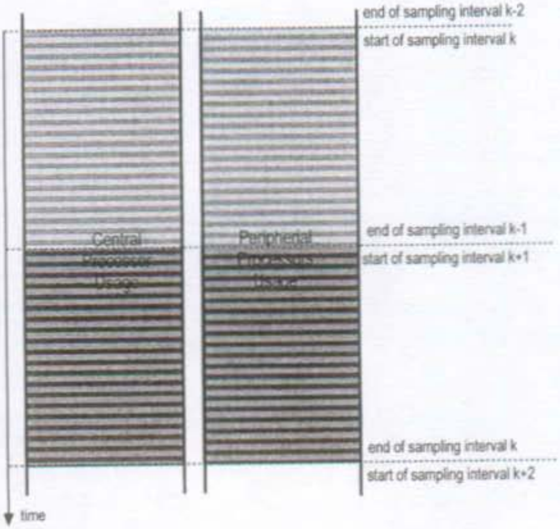

end of sampling interval k-2
start of sampling interval k
Central
Processor
Usage
Peripherial
Processors
Usage
end of sampling interval k-1
start of sampling interval k+1
end of sampling interval k
start of sampling interval k+2
time

Fig.24 : Interleaving Sampling Intervals to obtain 100% CPU Usage in a Two-Stages Pipelined Control Unit.

IV.7.3 Control Unit Design.

Practical Constraints and the Transputer.

From an implementation point of view, the design of such a control software is possible only when a *real-time kernel* features *pseudo-parallelism* on each processor, which is represented by the horizontal lines pattern in Fig. 24. Otherwise it would be an impossible task for the programmer to code manually the sequence of operations to run on each processor. The Transputer has the advantage of offering a hardware kernel with ultra-fast task switching that introduces almost no overhead.

An important overhead is nevertheless introduced by the communications between processors. Fig. 21 shows that not less than 9 exchange of data between the central and each of the peripherals must take place in one traversal of the algorithm, or 18 when pipeline is implemented. While the Transputer links are able to transfer large data packets at 20 Mbit/s, this rate falls to no more than 2 Mbit/s (about 60000 floating point numbers per second) when only a few bytes are transferred.

Finally, it is clearly not possible to obtain 50% business of each processor. The designer can be happy if rates between 40% and 60% can be obtained. After the implementation of the pipeline, an overall efficiency higher than 50% with $n = 3$ can be interpreted as a success since it means that the control unit works more than twice as fast as a single-processor control unit.

Description of an Experimental Transputer-Based Control Unit.

The experimental control unit developed for the model-based control of the Delta robot is sketched in Fig. 25. The computing power is provided by the four INMOS T-800 Transputers named "Central" and "Periph. 1-3".

Since the robot interface hardware is accessed through a single Transputer Link, a fifth "Host" T 800 had to be used. It also executes the development system (TDS) and simple tasks such as the user interface and the reference signal generation. This Transputer is much less loaded than the four others and would probably be replaced by the PC processor itself in a newer design.

The performance of this control unit is discussed in Chapter V, where the implementation of the various algorithms as well as the load-balancing procedure are detailed.

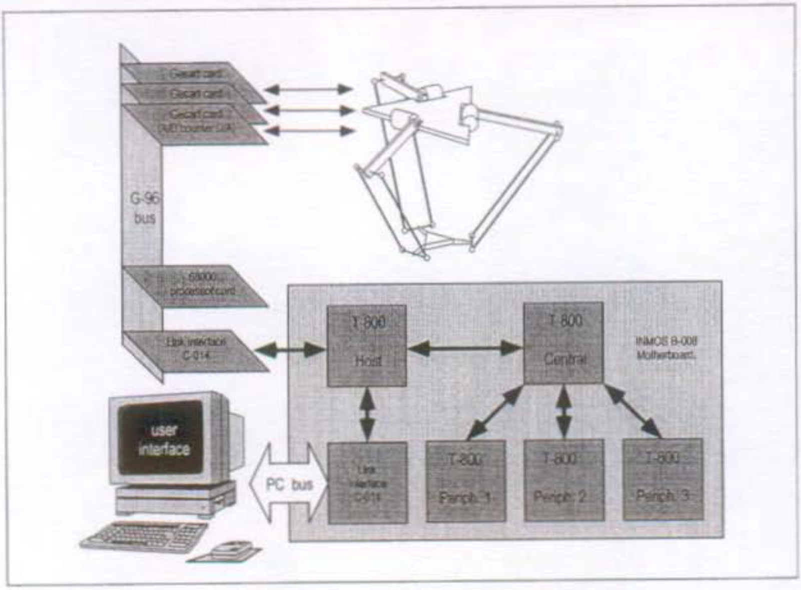


Fig.25 : Structure of the Experimental Control Unit.

IV.8 Conclusions.

A complete discussion of control techniques that were proposed for the control of robots leads to consider them as compromises between purely model-based and purely robust approaches. Experimental results showing the effectiveness of model-based strategies are more convincing than the available results regarding adaptive or robust control, especially when fast manipulators are considered.

Moreover, the advantages of task space control over joint space schemes are discussed. For parallel robots, a model-based scheme with a feedback loop in task space requires fewer computations than its equivalent in joint space. Such a scheme is therefore proposed, where the inverse dynamics model in the two spaces is involved.

Finally, the structure of the proposed algorithm is examined and an efficient implementation on a multiprocessor control unit with a parallel+pipeline structure is described.

Chapter V

Application to the Delta Robot.

V.1 Introduction.

In this chapter, the modeling process and controller design procedure proposed in this thesis are applied to the Delta robot. The kinematics and dynamics of the Delta robot as well as implementation issues are discussed in great detail.

In *Section V.2, the structure of the Delta manipulator and its application to fast pick & place tasks* are first presented. Under the hypothesis that the sub-chains are identical, the parametrization of the manipulator is discussed and the kinematics and dynamics parameters of the experimental Delta given.

The kinematics of the Delta is completely solved in Section V.3, where the inverse algorithms for positions, velocities and accelerations are detailed. A Newton-Raphson forward kinematics algorithm is shown to be more general than a closed-form formulation. Special attention is given to the analysis of the complexity of the proposed algorithms. It is shown that they all efficiently take advantage of a multiprocessor control unit.

The problem of the singularities and inverse singularities is dealt with in Section V.4. Their location is analyzed and a map of the Jacobian's determinant over the robot's workspace is given to show their influence in it. The very difficult problem of defining a usable workspace in which the Delta robot offers a bounded stiffness is considered. A numerical case study leads to an attempt of defining the usable workspace with respect to the inverse singularities.

In Section V.5, the dynamics of the Delta robot is analyzed. First, a simplified model where the inertia momentum of the forearms is neglected is presented in the format of a model in the two spaces. This simplifying hypothesis is then removed and a full model is obtained through a Lagrange approach where kinematics is explicitly introduced.

Some simulation and experimental results lead to a critical discussion on the Delta's performances and limitations in Section V.6.

The main contributions in this chapter are finally summarized in Section V.7.

V.2 Experimental Setup.

V.2.1 The Delta Manipulator.

The Delta is a 3-DOF fully parallel robot designed by Clavel at the Institut de Microtechnique (IMT) of the EPFL [Clav85], [Clav88], [Clav91]. The main objective was to design a very fast manipulator for pick & place tasks. Typically, it can pick a 10 g mass and place it 30 cm further along a half-elliptic trajectory in 0.12 [s], allowing a 4 cycles/s cadency. The platform can reach accelerations of 300 m/s^2 and travel speeds close to 10 [m/s]. These performances allow to say that the Delta robot is the fastest industrial manipulator in the world.

The Delta is operated upside-down, the base "hanged on the ceiling" as represented in Fig. 26.

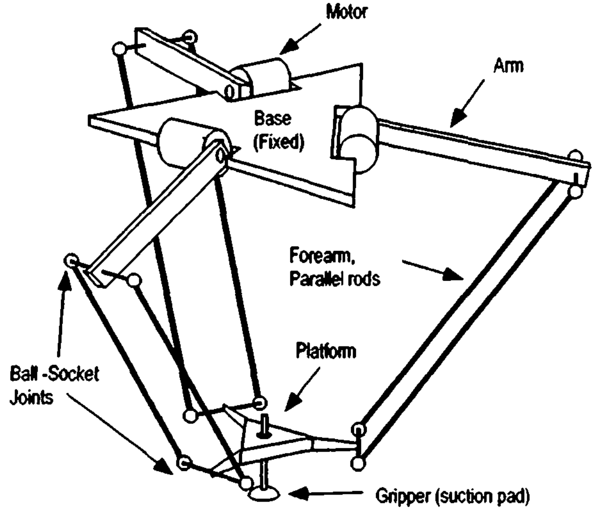


Fig.26 : The Delta Robot.

The Delta's structure has two imbricated types of closed kinematic chains:

- Three sub-chains connect the base to the platform. Each consists of two parts: an arm attached to the robot's base through an actuated 1-DOF revolute joint and a forearm linking the arm to the platform holding the end-effector.
- Each forearm is made of a pair of rods with ball-sockets joints at both edges. Struts maintain each pair of ball-socket joints at each edge of the forearm separated. The three constraints introduced by these three rigid closed chains remove three degrees of freedom of the platform. In particular, when the structure has certain symmetry properties, the forearms form what Clavel calls a "parallelogram in space" that prevents the platform's rotations. The Delta is then functionally equivalent to a 3-DOF Cartesian carrier.

V.2.2 Geometrical Description.

Standard Parametrization of the 6-DOF Structure.

According to the parametrization of parallel robots introduced in § II.7.5, the Delta structure is described by the Denavit-Hartenberg parameters of its base tree (see Table 7) and upper tree (see Table 8).

i	a_i	γ_i	ζ_i	d_i	α_i	θ_i	r_i
1	0	0	0	r_a	90°	q_1	0
2	0	120°	0	r_a	90°	q_2	0
3	0	240°	0	r_a	90°	q_3	0
4	1	0	0	l_a	180°	-	l_e
5	2	0	0	l_a	180°	-	l_e
6	3	0	0	l_a	180°	-	l_e
7	1	0	0	l_a	180°	-	$-l_e$
8	2	0	0	l_a	180°	-	$-l_e$
9	3	0	0	l_a	180°	-	$-l_e$

Table 7 : Parametrization of the Delta Manipulator Base Tree.

Moreover, the signed length of the six links is required. Each forearm can be viewed as a complex link of type 0 (see Table 2). The internal degrees-of-freedom are removed by springs that connect each pair of rods. Therefore, ${}^{1}l = \begin{bmatrix} l_b & l_b & l_b & -l_b & -l_b & -l_b \end{bmatrix}^T$.

i	a_i	γ_i	ζ_i	d_i	α_i	θ_i	r_i
1	0	0	0	r_b	-	-	l_e
2	0	120°	0	r_b	-	-	l_e
3	0	240°	0	r_b	-	-	l_e
4	0	0	0	r_b	-	-	$-l_e$
5	0	120°	0	r_b	-	-	$-l_e$
6	0	240°	0	r_b	-	-	$-l_e$

Table 8 : Parametrization of the Delta Manipulator Upper Tree.

Simplifying Hypothesis and 3-DOF Parametrization.

The geometry of the Delta robot is sketched in Fig. 27.When the three sub-chains are identical and the three-fold symmetry perfect, five scalar parameters $\{r_a, r_b, l_a, l_b, l_e\}$ suffice to specify the dimensions of the whole structure.

Since the platform remains parallel to the base, only $r_r = r_a - r_b$ appears in the kinematics. l_e and r_b/r_a are not involved in kinematics but guarantee the stiffness of the structure. Each forearm can therefore be modeled as a single bar of length l_b.

V.2.3 The Delta-IA Robot.

All the numerical and experimental results presented in this chapter were developed for a prototype of the Delta robot designed at IMT and built in collaboration with the Institut d'Automatique (IA) of EPFL where it is used as a test bench for robot control algorithms.

Sensors, Actuators and Transmissions.

Three brushless DC-motors offering 2 [Nm] maximum torque drive each arm through a crank belt transmission with $n_r = 10 \div 1$ reduction ratio. Incremental sensors with 20'000 increments per turn are mounted on the motors.

For the sake of simplicity, the experimental case study proposed in this chapter and the control software were developed for a fictitious equivalent direct-drive robot with actuators offering 20 [Nm] and sensors with 200'000 increments per arm turn.

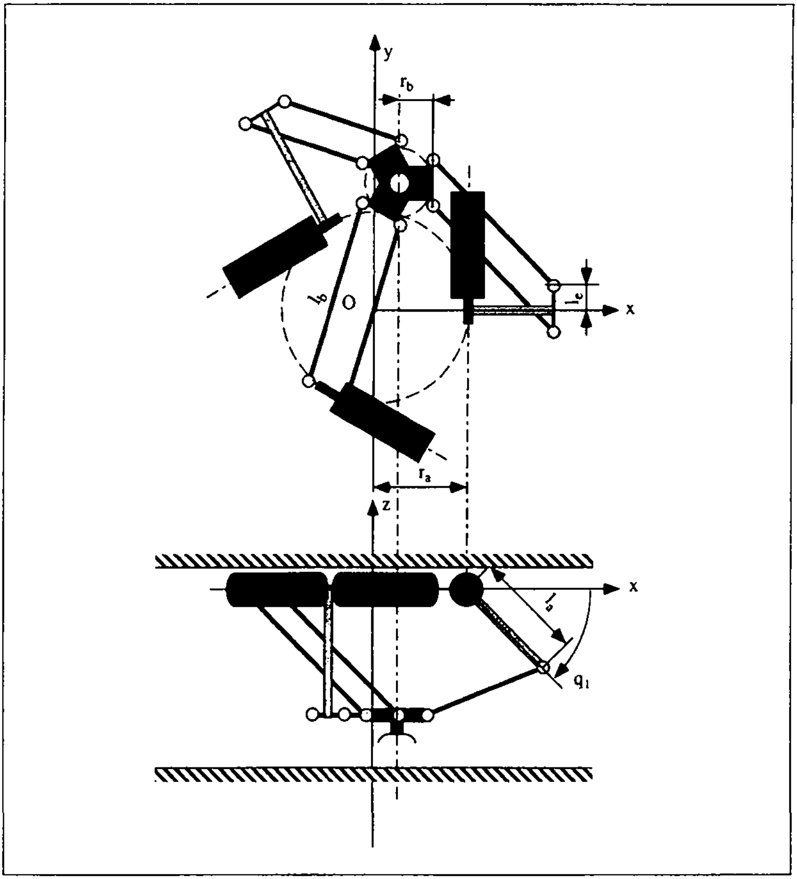


Fig.27 : Geometry of the Delta Robot.

Geometrical Dimensions.

The Delta-IA has the following dimensions:

$$\begin{cases} l_a = 0.260\,[\mathrm{m}] \\ l_b = 0.480\,[\mathrm{m}] \\ r_a = 0.194\,[\mathrm{m}] \\ r_b = 0.030\,[\mathrm{m}] \end{cases} \qquad \text{(V.1)}$$

According to Clavel's convention, it is a Delta 740 since $l_a + l_b = 740\,[\text{mm}]$. These geometrical dimensions result from Clavel's study on their relation with the robot's workspace size [Clav91]. For pick & place tasks, a good compromise between the workspace's height and surface is obtained with

$$r_r/l_a \cong 0.63 \text{ and } l_b/l_a \cong 2. \qquad (V.2)$$

Dynamics Parameters.

Extending the hypothesis of perfect geometrical symmetry of the robot (see § V.2.2) to dynamics, only a handful of parameters are required to describe the inertia of each body. Because the Delta's structure can easily be disassembled, it is easy to accurately measure the masses m_a of each arm, m_b *of each pair of the rods (forearm) and* m_c *of the platform:*

$$\begin{cases} m_a = 0.200[\text{Kg}] \\ m_b = 0.088[\text{Kg}] \\ m_c = 0.092[\text{Kg}] \end{cases}. \qquad (V.3)$$

The arms and forearms where considered as homogeneous bodies with an axial symmetry, their center of gravity laying in their middle:

$$\begin{cases} g_a = l_a/2 = 0.130\,[\text{'m}] \\ g_b = l_b/2 = 0.240\,[\text{'m}] \end{cases}. \qquad (V.4)$$

With the same hypothesis, the inertia momentum of the forearms about any axis orthogonal to the rod passing through the center of gravity is

$$j_b = \frac{m_b l_b^2}{12} = 0.0017\,[\text{Kg.m}^2]. \qquad (V.5)$$

Finally, the total inertia j_a of the arm, *actuator rotor and transmission was identified through an off-line* least squares identification procedure (see § V.6.3). The obtained value is

$$j_a = 0.0131\,[\text{Kg.m}^2]. \qquad (V.6)$$

V.3 Kinematics.

V.3.1 Preliminary Definitions.

To simplify the following developments, three auxiliary cartesian frames are obtained by rotating the reference frame around the z axis according to the rotation matrices:

$$ {}^{i}R = \begin{bmatrix} \cos\gamma_i & -\sin\gamma_i & 0 \\ \sin\gamma_i & \cos\gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, i = 1, 2, 3, \qquad \text{(V.7)} $$

where the γ_i angles are taken from Table 7. A unit vector along each arm is defined in the corresponding auxiliary frame:

$$ {}^{i}v = \begin{bmatrix} \cos q_i \\ 0 \\ \sin q_i \end{bmatrix}. \qquad \text{(V.8)} $$

It is useful to define the V matrix as the basis generated by these unit vectors in the operational space:

$$ V = \left[{}^{1}R \cdot {}^{1}v \middle| {}^{2}R \cdot {}^{2}v \middle| {}^{3}R \cdot {}^{3}v \right]. \qquad \text{(V.9)} $$

Then, the position of the three first connection points on the base tree is

$$ {}^{b}P_i = {}^{i}R \cdot \begin{bmatrix} r_a \\ l_e \\ 0 \end{bmatrix} + l_a V_i, i = 1, 2, 3. \qquad \text{(V.10)} $$

The three corresponding connection points on the upper tree, which is the platform itself, are

$$ {}^{u}P_i = p + {}^{i}R \cdot \begin{bmatrix} r_b \\ l_e \\ 0 \end{bmatrix} = {}^{i}R \cdot \left({}^{i}p + \begin{bmatrix} r_b \\ l_e \\ 0 \end{bmatrix} \right), \; i = 1, 2, 3, \tag{V.11} $$

where

$$ {}^{i}p = {}^{i}R^{-1} \cdot p = {}^{i}R^{T} \cdot p \tag{V.12} $$

since ${}^{i}R$ is an orthogonal matrix.

The constraint introduced by the corresponding link is

$$ {}^{i}w^{T} \cdot {}^{i}w - 1 = 0, \tag{V.13} $$

where

$$ {}^{i}w = \frac{1}{l_b} {}^{i}R^{T} \cdot \left[{}^{u}P_i - {}^{b}P_i \right] = \frac{1}{l_b} \left({}^{i}p - \begin{bmatrix} r_r \\ 0 \\ 0 \end{bmatrix} - l_a {}^{i}v \right). \tag{V.14} $$

is the unit vector along the ith forearm in its corresponding auxiliary frame and

$$ W = \left[{}^{1}R \cdot {}^{1}w \,\middle|\, {}^{2}R \cdot {}^{2}w \,\middle|\, {}^{3}R \cdot {}^{3}w \right] \tag{V.15} $$

the basis generated by these unit vectors in operational space.

V.3.2 Inverse Kinematics.

Many different approaches are possible to obtain an inverse kinematics algorithm. However, special attention should be given to singularities that can artificially be introduced. Sternheim's algorithm [Ster87] and Clavel's both suffer from such defects, that were solved by Codourey [Codo91a]. Here, a corrected and optimized version of this latter algorithm is proposed.

The i^{th} constraint equation (Eq. V.13) can be written as

$$\left({}^ip_x - r_r\right){}^iv_x + {}^ip_z\,{}^iv_z - l_c = 0 \tag{V.16}$$

where

$$l_c = \left(l_a^2 - l_b^2 + \left({}^ip_x - r_r\right)^2 + {}^ip_y^2 + {}^ip_z^2\right)/2l_a . \tag{V.17}$$

From Eq. V.16,

$${}^iv_z = \left(l_c - \left({}^ip_x - r_r\right){}^iv_x\right)/{}^ip_z \tag{V.18}$$

Since ${}^iv$ is a unit vector with ${}^iv_y = 0$

$${}^iv_x^2 + {}^iv_z^2 = 1 . \tag{V.19}$$

Combining Eq. V.17 and Eq. V.19 yields a second order equation in ${}^iv_x$, with solutions

$${}^iv_x = \frac{l_c\left({}^ip_x - r_r\right) \pm {}^ip_z\sqrt{{}^ip_z^2 + \left({}^ip_x - r_r\right)^2 - l_c^2}}{{}^ip_z^2 + \left({}^ip_x - r_r\right)^2} . \tag{V.20}$$

The solution with the plus sign is excluded since it corresponds to a situation where the arm is directed towards the inside of the robot. Finally, from Eq. V.8,

$$q_i = \operatorname{atan2}({}^iv_z, {}^iv_x) \tag{V.21}$$

where atan2 is a function of two parameters that computes an arctangent in the four quadrants. It is now commonly available in numerical libraries.

Complexity Analysis.

This algorithm requires 69 FLOP plus the expensive evaluation of three square root and three atan2. On a single T-800 Transputer processor, it requires 240 [µs]. This is 12 FLOP or 4 [µs] less than Codourey's algorithm [Codo91a]. When implemented on the three "peripheral" T-800 working in parallel, the computation time drops to 88 [µs], which corresponds to a very satisfactory 91% efficiency.

V.3.3 Inverse Jacobian.

Differentiating Eq. V.14 with respect to time leads to a relation between the joint velocities and platform speeds in the auxiliary frames

$$ {}^{i}\dot{w} = \frac{1}{l_b} {}^{i}R^{T} \cdot \left[{}^{u}\dot{\vec{P}}_i - {}^{b}\dot{P}_i \right] = \frac{1}{l_b}\left({}^{i}\dot{p} - l_a {}^{i}\dot{v} \right) = \frac{1}{l_b}\left({}^{i}\dot{p} - l_a {}^{i\perp}v\dot{q}_i \right), \; i = 1, 2, 3, \tag{V.22} $$

where

$$ {}^{i\perp}v = \frac{\partial {}^{i}v}{\partial q_i} = \begin{bmatrix} -\sin q_i \\ 0 \\ \cos q_i \end{bmatrix}. \tag{V.23} $$

After some transformations detailed in [Gugl91], Eq. V.22 for $i = 1, 2, 3$ can be written in the form:

$$ W^{T} \cdot \dot{p} = l_a X \cdot \dot{q}, \tag{V.24} $$

where

$$ X = \begin{bmatrix} {}^{1}w^{T} \cdot {}^{1\perp}v & 0 & 0 \\ 0 & {}^{2}w^{T} \cdot {}^{2\perp}v & 0 \\ 0 & 0 & {}^{3}w^{T} \cdot {}^{3\perp}v \end{bmatrix}. \tag{V.25} $$

Considering Eq. V.24 with respect to Eq. II.17, one can see that the Delta's inverse Jacobian is given by a rather simple expression:

$$ J^{-1} = \frac{1}{l_a} X^{-1} \cdot W^{T}, \tag{V.26} $$

while, as explained in § II.9.6, the closed form of the Jacobian

$$ J = l_a X \cdot W^{-T} \tag{V.27} $$

is too complicated to be of any practical use since an inversion of W is involved.

Complexity Analysis.

The inverse kinematics algorithm described in § V.3.2 internally computes the ${}^{i}v$ vectors required for the inverse Jacobian evaluation. Obtaining the ${}^{i}w$ vectors from Eq. V.14 and the inverse Jacobian (Eq. V.15, Eq. V.25 and Eq. V.26) then requires only 60 more FLOP. This takes 140 [μs] on a single T-800. On three T-800, inverse kinematics and inverse Jacobian are computed in only 128 [μs]. Efficiency is then as high as 96% because more computations are performed in parallel for the same communication overhead.

Then, a straightforward numerical inversion of the inverse Jacobian would require 45 FLOP (65 [μs]). In the proposed control scheme (see Fig. 19), the Jacobian is only used in dot products with vectors, which each requires 15 FLOP.

Since such an operation is equivalent to solving a linear system of equation where the inverse Jacobian is the system's matrix, another approach is to LU-decompose J^{-1}, which requires only 13 FLOP (24 [μs]), and then spend 15 FLOP to solve each system. This technique, which saves only 40 [μs] in the case of the Delta, leads to very important savings in the case of 6-DOF manipulators.

V.3.4 Forward Kinematics and State Observer.

Why the Closed-Form Forward Algorithm Should Not Be Used.

Forward kinematics requires to solve Eq. V.13 simultaneously for $i = 1, 2, 3$, which is a system of quadratic equations in p. Because of the very particular structure of the perfect Delta, this problem is geometrically equivalent to finding the intersection of three spheres, for which a closed-form solution exists [Clav88], [Codo91a]. As implemented by Codourey, this takes 268 [μs] on a single T-800.

Researchers should be aware that the condition of existence of such a closed-form is very delicate. A single change in one of the robot's parameters is sufficient to destroy it. All the results that suppose the existence of a closed-form forward kinematics relation cannot be generalized yet since such a relation is not currently available for a general parallel manipulator. For theis reason, it is advised here not to use the forward kinematics relation any more.

Use the Iterative Newton-Raphson Algorithm Instead!

The forward kinematics transformation required in the state observer block of the proposed control scheme (see Fig. 19) is implemented using the Newton-Raphson algorithm described in § II.9.6. A measure of the convergence rate of this algorithm is the number of iterations n_s required to achieve a precision corresponding to one increment of the robot's position sensors. Since n_s depends on the quality of the first approximation, the worst case is obtained when the robot moves at its maximum velocity.

A simulation was performed where the robot is supposed to move at 10 [m/s] all over the workspace. Considering a sampling frequency of 1 [KHz], successive points in operational space are 1 [cm] apart. The complete algorithm with Jacobian update requires exactly two iterations to reach the desired precision but takes 928 [µs] of a single T-800 CPU time. When the Jacobian is not re-evaluated only once, three steps are required in less than 2% of the cases, and CPU time is reduced to 640 [µs]. This might seem very slow compared to the closed-form forward kinematics algorithm.

However, fixing the number of iterations to $n_s = 2$ allows to optimize the iterative algorithm and exploit its parallel+pipeline structure on four processors. Since the inverse Jacobian and its LU-decomposition are evaluated in the algorithm, the direct speed transformation $\dot{p} = J \cdot \dot{q}$ can be performed at a very low extra cost. Finally, the complete state observer requires 343 [µs] on four T-800 processors. The efficiency is close to 50%, indicating that the computational load is equally divided between the central processor and the three peripheral processors.

V.3.5 Inverse Jacobian's Time Derivative and the Acceleration Mapper.

$\dot{J}^{-1}$ is obtained by differentiating Eq. V.26 with respect to time:

$$\dot{J}^{-1} = \frac{1}{l_a} X^{-2} \cdot \left[X \cdot \dot{W}^T - \dot{X} \cdot W^T\right], \tag{V.28}$$

where

$$\dot{X}_{i,i} = {}^{i}\dot{w}^T \cdot {}^{i\perp}v - {}^{i}w^T \cdot {}^{i}v\dot{q}_i \tag{V.29}$$

since ${}^{i\perp}\dot{v} = -{}^{i}v\dot{q}_i$ (see Eq. V.8 and Eq. V.23), and

$$\dot{W} = \left[{}^1R \cdot {}^1\dot{w} \middle| {}^2R \cdot {}^2\dot{w} \middle| {}^3R \cdot {}^3\dot{w}\right]. \tag{V.30}$$

The power of the approach in the two spaces appears since Eq. V.22 allows to compute the required ${}^{i}\dot{w}$ vectors once the velocities are available in both operational and joint spaces.

In the proposed control scheme, $\dot{J}^{-1}$ is only used to map operational accelerations to joint space thanks to Eq. II.31, which is reminded here:

$$\ddot{q} = J^{-1} \cdot \ddot{p} + \dot{J}^{-1} \cdot \dot{p}. \tag{V.31}$$

Complexity Analysis.

Evaluating $J^{-1} \cdot \dot{p}$ takes 120 FLOP or 268 [μs] of T-800 CPU time. This computation can be completely parallelized on the three peripheral processors with a very high efficiency of 95%. Then, the acceleration mapper needs only 18 more FLOP to complete.

The kinematics part of the control scheme proposed in Fig. 19 finally takes 1320 [μs] on a single T–800 and 582 [μs] when parallelized, giving a 57% efficiency. Considering the FLOP count and the communication overhead, this reveals that the peripherals processors are more loaded than the central processor.

V.4 Singularities and Definition of the Usable Workspace.

V.4.1 Singularities and Inverse Singularities.

The geometrical meaning of W (Eq. V.15) and X (Eq. V.25) helps to find the configurations where either the Jacobian or its inverse becomes singular.

- Elements on the diagonal of X contain the cosine of the angle between ${}^{i}w$ along the forearm and ${}^{i\perp}v$, which is orthogonal to the arm and belongs to its plane of rotation. Therefore, the Jacobian (Eq. V.27) becomes singular when at least one forearm is colinear to the arm of the same sub-chain. This occurs only on the external surface of the robot's workspace.
- Columns of W contain the unit vectors along the forearms in the same reference frame. This matrix and J^{-1} (Eq. V.26) become singular when two or more forearms are parallel or coplanar.

Inverse singularities may exist inside the robot's workspace. However, designing a Delta such that $l_b > l_a + r_r$ makes it impossible to have the three forearms in the same plane, and this is guaranteed by Clavel's first design rule (Eq. V.2). To forbid configurations where forearms are parallel, r_r should be larger than l_a, which is not the case according to the second design rules. Consequently, a subspace of the workspace should be defined, where the robot can be operated safely.

Analysis of the Jacobian's Determinant.

As mentioned earlier in this work, a parallel robot has advantages over a serial arm only if it is operated "far enough" from the singularities of both types. However, no physically meaningful norm available as a measure of the distance to them. In a first attempt to qualitatively represent the regularity of the kinematics in the workspace, the Jacobian's determinant might be used.

Fig. 28 represents the value of $|J|$ on seven horizontal slices of the Delta's workspace. In the upper part of the workspace, the proximity of three inverse singularities clearly appears. The three levels in the middle correspond to the zone where the robot is usually operated in pick & place tasks. The robot's

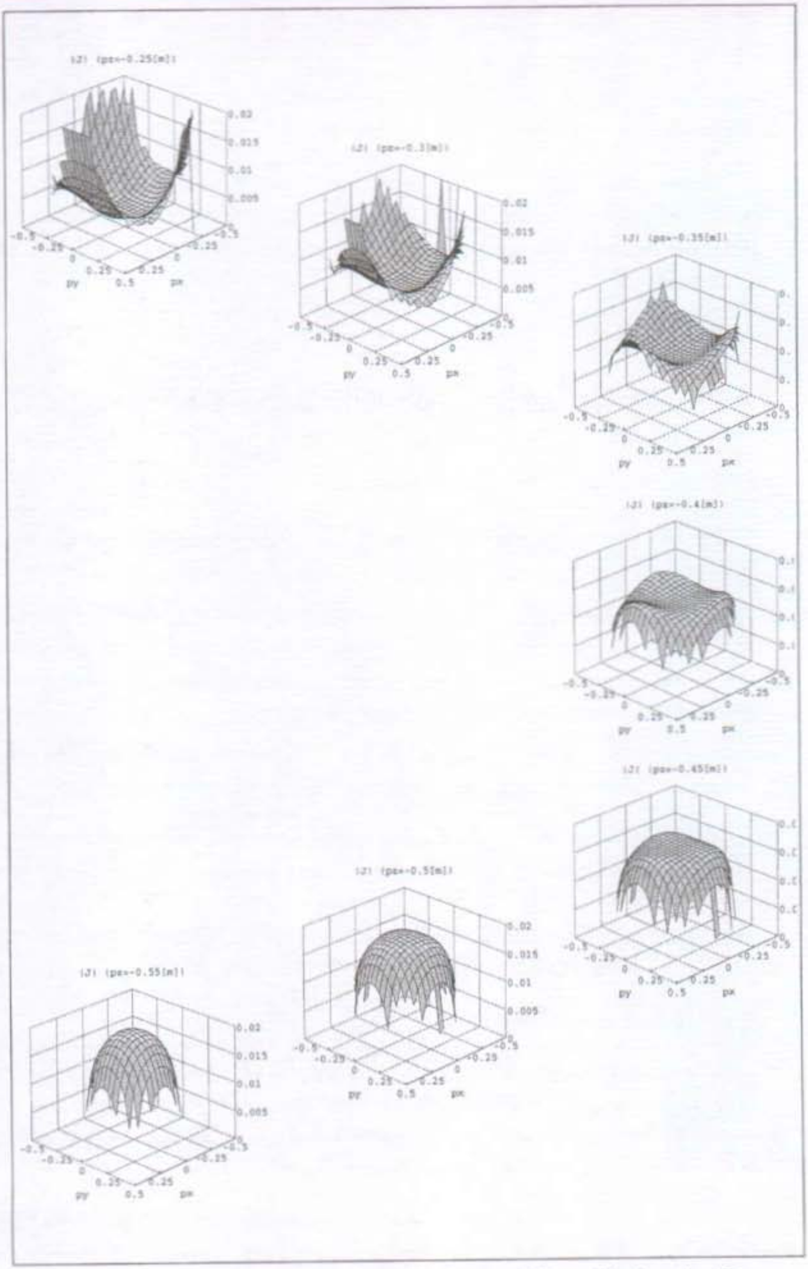


Fig.28 : Jacobian's Determinant on Seven Equidistant Horizontal Slices of the Operational Space.

properties are here remarkably constant over a wide part of the workspace. Only three small bumps reveal the influence of the inverse singularities. At the lower levels, the curved surface indicates the neighborhood of another inverse singularity in the center of the section.

V.4.2 Defining a Usable Workspace.

A Purely Geometrical Definition.

Clavel based the original definition of the Delta's usable workspace on mechanical design considerations. Limits were posed on the extent of some angles to prevent collisions between bodies and avoid dislocation of the ball-socket joints. This definition allows to obtain an approximation of the workspace's volume and main dimensions. The possibility of entering inverse singularities is avoided thanks to additional constraints, but security margin is introduced. Moreover, the algorithm to check if a point belongs to the workspace requires to check the values of the angles at the free joints. It can therefore not easily be combined with the kinematics algorithm presented in this chapter. In practice, existing implementation simply check that the robot's position in joint space belongs to the cube

$$q_{min} \le q_i \le q_{max}, \; i = 1, 2, 3 \tag{V.32}$$

where q_{min} and q_{max} are chosen such that the other constraints are satisfied.

A New Definition based on Performance Considerations.

It is proposed here to consider a new definition of the workspace $\mathcal{W}$ as the subspace of O where eigenvalues of the matrix $J^{-T} \cdot J^{-1}$ stay within given bounds $\bar{\beta}$ and $\underline{\beta}$:

$$\mathcal{W} = \{ p \in O | \underline{\beta} \le \mathrm{eig}[J^{-T}(p) \cdot J^{-1}(p)] \le \bar{\beta} \} . \tag{V.33}$$

The physical meaning of this definition can be understood when considering Eq. III.5 in § III.3.1. For a fully parallel robot like the Delta, all the $k_{e,i}$ elasticity constants can be considered equal and the $J^{-T} \cdot J^{-1}$ matrix appears as a factor in the stiffness matrix. The bounds $\bar{\beta}$ and $\underline{\beta}$ therefore limit the workspace to the configurations where the structure has a sufficient but finite stiffness. Singularities of the Jacobian and inverse singularities are therefore excluded with a physically meaningful security margin. Note that only static stiffness is considered: natural frequencies can not easily be determined because of the varying inertia and couplings.

However the proposed definition can clearly not be easily implemented. A solution is to consider the simple definition from Eq. V.32 and to determine numerically the bounds q_{min} and q_{max} such that the corresponding workspace is included in $\mathcal{W}$.

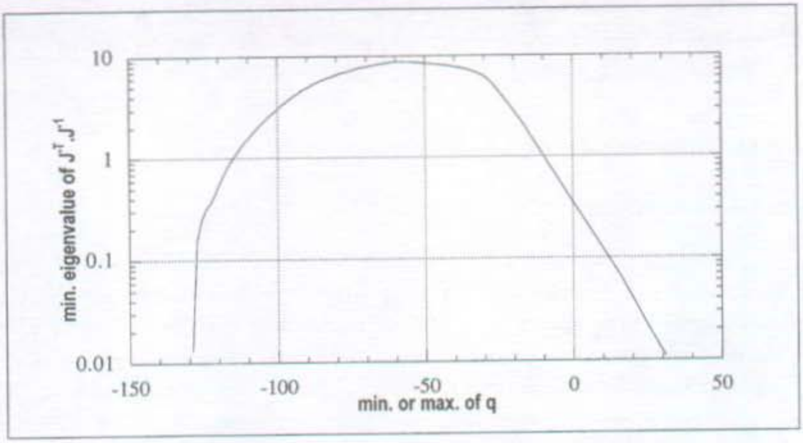


Fig.29 : Minimum eigenvalue of $J^{-T} . J^{-1}$ as a function of the bounds of q .

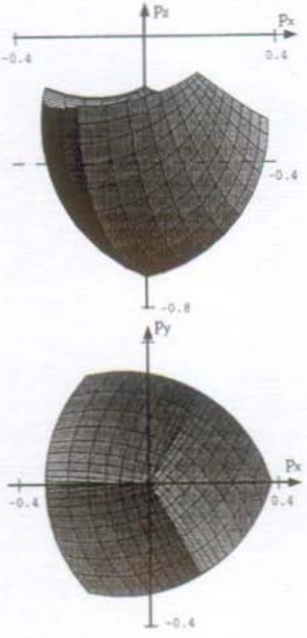


Fig.30 : Usable Workspace of the Delta 740.

The lowest eigenvalue of $J^{-T} \cdot J^{-1}$ attained in the workspace is plotted in Fig. 29 as a function of q_{min} in the right hand side of the graph and q_{max} in its left hand side. Note how the stiffness decreases rapidly when $q_{min} \rightarrow \text{acos}\,(-r_r/l_a) \cong -129°$ but slower as q_{max} increases. On the other hand, it can be seen that the closed chain structure of the Delta results in stiffening the open chain mechanism up to a factor 9 for some configurations.

Choice of the Bounds.

Choosing $\underline{\beta} = 1$ guarantees that the robot is stiffer than an serial arm with the same joint elasticity, but the corresponding workspace is then very much reduced.

Considering the low inertia of the open chains and the high associated natural frequency (about 88 Hz according to the identification process detailed in § V.6.3), choosing $\underline{\beta} = 0.1$ still ensures correct performance of the robot.The corresponding workspace, given by $q_{min} = -125°$ and $q_{max} = 15°$ is *represented in Fig. 30.*

V.5 Dynamics.

Many different techniques were used in various attempts to obtain an inverse dynamics model of the Delta. In early attempts, no distinction was made between kinematics and dynamics. The resulting models were extremely complicated and therefore not usable for analysis or real-time implementation.

Another approach was initiated by Codourey, where a much simpler model was developed, that could be used in a model-based control strategy [Codo91a]. The model's formulation was greatly simplified by assuming forearms with negligible inertia momentum. This hypothesis was then removed by Miller, who obtained a relatively simple model through a Lagrange-with-multipliers approach [Mill92]. In both works however, the difficulty is somewhat alleviated by using numerical differentiation techniques to obtain the robot's state in both spaces instead of exact kinematics relations.

In this context, the contribution of this thesis is to introduce kinematics in the dynamics model, allowing to replace numerical differentiations by explicit expressions. This approach furthermore allows to obtain models equivalent to Codourey's and Miller's, but with a much simpler formulation that allows analysis of the robot's dynamics.

The complete inverse dynamics model the Delta is established in three steps using a Lagrange based approach. First, only gravity forces are considered. Then a simplified model is obtained, where no Coriolis forces appear. Finally, the complete model in the two spaces of the Delta is given and discussed.

V.5.1 Modelling Gravity Effects.

The potential energy of the Delta in an uniform potential field defined by the vector ς is:

$$e_P = \varsigma^T \cdot \left[m_c p + m_a g_a V^T \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + m_b \left[l_a V^T + g_b W^T \right] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right] . \tag{V.34}$$

In the usual case where the z axis of the robot is aligned with the Earth's gravity field, $\varsigma^T = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}$ [m/s²]. Eq. V.34 becomes

$$e_P = 9.81 \left(m_c + 3 \frac{m_b g_b}{l_b} \right) p_z + 9.81 \sum_{i=1}^{3} \left(m_a g_a + m_b l_a \left(1 - g_b / l_b \right) \right) {}^i v_z . \tag{V.35}$$

From this expression, the gravity forces vectors of the model in the two spaces Eq. III.27 are easily obtained:

$${}^q g(q) = 9.81 \left(m_a g_a + m_b l_a \left(1 - \frac{g_b}{l_b} \right) \right) \begin{bmatrix} \cos q_1 & \cos q_2 & \cos q_3 \end{bmatrix}^T \tag{V.36}$$

and

$${}^p g(p) = \left[0 \;\; 0 \;\; 9.81 \left(m_c + 3 \frac{m_b g_b}{l_b} \right) \right]^T . \tag{V.37}$$

As mentioned in § IV.2.2, PD+gravity controllers are the simplest for which formal proofs of convergence exist. In the case of the Delta, *the control law in joint space corresponding to Eq. IV.7 is*

$$u = K_p \tilde{q} + K_d \dot{\tilde{q}} + {}^q g(q) + J^T \cdot {}^p g(p) . \tag{V.38}$$

V.5.2 The Simplified Model.

Modelling the forearms as two punctual masses connected by rods of negligible inertia momentum, the kinetic energy of the manipulator is

$$e_K = \frac{1}{2}\left(m_c + 3m_b\frac{g_b}{l_b}\right)\|\dot{p}\|^2 + \frac{3}{2}\left(j_a + m_b l_a^2\left(1 - \frac{g_b}{l_b}\right)\right)\|\dot{q}\|^2, \tag{V.39}$$

where g_b/l_b is the ratio of the forearm's mass placed at the connection point with the platform.

The inertia matrix in generalized coordinates can immediately be extracted from Eq. V.39. Since it is diagonal, it has the more general block diagonal structure given in Eq. III.26 and its elements are constant, i.e. they do not depend on the manipulator's position:

$${}^{q}A(q) = \left(j_a + m_b l_a^2\left(1 - \frac{g_b}{l_b}\right)\right)I, \tag{V.40}$$

$${}^{p}A(p) = \left(m_c + 3m_b\frac{g_b}{l_b}\right)I. \tag{V.41}$$

Consequently, the corresponding quadratic forces matrices ${}^{q}C(q, \dot{q})$ and ${}^{p}C(p, \dot{p})$ are zero. The simplified inverse dynamics model in the two spaces therefore is

$$\tau = {}^{q}A(q)\cdot\ddot{q} + {}^{q}g(q) + J^T\cdot\left[{}^{p}A(p)\cdot\ddot{p} + {}^{p}g(p)\right]. \tag{V.42}$$

Note that some quadratic forces are indeed modeled in Eq. V.42 because the acceleration mapper involves products of velocities (see Eq. II.31 and Eq. II.32). Intuitively, when the platform follows a curved trajectory, its centripetal acceleration is mapped onto joint space.

V.5.3 The Complete Model in the Two Spaces.

Miller established the first complete model of the Delta through a Lagrange-with-multipliers approach [Mill92]. It could therefore not be put in a form that allows analysis because the implicit inertia matrix in generalized coordinates has not the block diagonal structure of Eq. III.26 since couplings remain between joint coordinates and operational coordinates. As shown below, these couplings can be removed, so that an equivalent model in the two spaces with the form of Eq. III.27 is obtained.

The kinetic energy of the i^th^ forearm is

$$
\begin{aligned}
{}^{i}e_K &= \frac{1}{2}\left(m_b \left| \frac{{}^{u}\dot{P}_i + {}^{b}\dot{P}_i}{2} \right|^2 + j_b \left| \frac{{}^{u}\dot{P}_i - {}^{b}\dot{P}_i}{l_b} \right|^2 \right) \\
&= \frac{1}{2}\left(\left(\frac{m_b}{4} + \frac{j_b}{l_b^2} \right) |\dot{p}|^2 + \left(\frac{m_b}{4} + \frac{j_b}{l_b^2} \right) \left\| \left({}^{i}\dot{p} - l_a\, {}^{i\perp}v \dot{q}_i \right) \right\|^2 + \left(\frac{m_b}{2} - 2\frac{j_b}{l_b^2} \right) \left({}^{i}\dot{p}^T \cdot \left({}^{i}\dot{p} - l_a\, {}^{i\perp}v \dot{q}_i \right) \right) \right) \\
&= \frac{1}{2}\left(m_b |\dot{p}|^2 + \left(\frac{m_b}{4} + \frac{j_b}{l_b^2} \right) l_a^2 \dot{q}_i^2 - m_b l_a \dot{p}^T \cdot {}^{\perp}V_i \dot{q}_i \right)
\end{aligned}
\tag{V.43}
$$

where

$$
{}^{\perp}V = \left[{}^{1}R \cdot {}^{1\perp}v \,\middle|\, {}^{2}R \cdot {}^{2\perp}v \,\middle|\, {}^{3}R \cdot {}^{3\perp}v \right] . \tag{V.44}
$$

The total kinetic energy of the system becomes

$$
e_K = \frac{1}{2}\left(\dot{p}^T \cdot \left[\left(m_c + \frac{3m_b}{2} \right) I - \frac{m_b l_a}{2} \left[{}^{\perp}V \cdot J^{-1} + J^{-T} \cdot {}^{\perp}V^T \right] \right] \cdot \dot{p} + \left(j_a + \left(\frac{m_b}{4} + \frac{j_b}{l_b^2} \right) l_a^2 \right) |\dot{q}|^2 \right) . \tag{V.45}
$$

The corresponding inertia matrix in joint space is diagonal and constant as in the simplified model:

$$
{}^{q}A(q) = \left(j_a + \left(\frac{m_b}{4} + \frac{j_b}{l_b^2} \right) l_a^2 \right) I , \tag{V.46}
$$

while the inertia matrix in operational space is here position dependent:

$$
{}^{p}A(p) = \left(m_c + \frac{3m_b}{2} \right) I - \frac{m_b l_a}{2} \left[{}^{\perp}V \cdot J^{-1} + J^{-T} \cdot {}^{\perp}V^T \right] . \tag{V.47}
$$

The quadratic forces matrix C might be obtained from the partial derivatives of e_K as described in § III.4.1. An alternative approach suggested by Slotine & Li [Slot91] is used here, which relies on the skew-symmetry of $\dot{A} - 2C$. As a consequence of this property:

$$
\dot{A} = C^T + C . \tag{V.48}
$$

Slotine & Li suggest that the C matrix can be guessed by examining $\dot{A}$ in the case of simple manipulators with a few degrees of freedom. However, it should be stressed here that Eq. V.48 alone does not define C in an unique manner. Therefore, one should be careful when using this technique and check that the guessed C is equal to the quadratic forces matrix obtained through Eq. III.13.

In the case of the Delta, this approach leads to ${}^{q}C = 0$ and

$$ {}^{p}C(p, \dot{p}, q, \dot{q}) = -\frac{m_b l_a}{2}\left[{}^{\perp}\dot{V} \cdot J^{-1} + {}^{\perp}V \cdot \dot{J}^{-1}\right]. \tag{V.49} $$

Note that velocities in joint space are still involved in Eq. V.49. They can easily be replaced by their equivalent in operational space using the inverse Jacobian if required. For the purpose of model based control, the formulation in Eq. V.49 is entirely satisfactory and the complete model in both spaces is:

$$ \tau = {}^{q}A(q) \cdot \ddot{q} + {}^{q}g(q) + J^{T} \cdot \left[{}^{p}A(p) \cdot \ddot{p} + {}^{p}C(p, \dot{p}, q, \dot{q}) \cdot \dot{p} + {}^{p}g(p)\right]. \tag{V.50} $$

V.5.4 Complete Versus Simplified Model: A Comparison.

As mentioned by Codourey, the assumption of negligible inertia momentum of the forearms is acceptable in practice when these parts are made of light material such as carbon fiber. In results below, the torques obtained from the simplified model differ only by a few percents of those obtained from a full model, and this happens only at very high velocities.

Codourey showed that the g_b parameter in the simplified model (Eq. V.40 and Eq. V.41) could be adjusted to minimize the tracking error. The best results were obtained with $g_b/l_b \cong 2/3$ [Codo91a]. Considering the complete model, this is justified since the introduction of Eq. V.5 in Eq. V.46 gives exactly Eq. V.40 while the inertia matrix in Eq. V.47 has eigenvalues close to those of Eq. V.41 in the central part of the workspace, where

$$ {}^{\perp}V \cdot J^{-1} + J^{-T} \cdot {}^{\perp}V^{T} \cong \frac{2}{l_a} I \tag{V.51} $$

since the vectors ${}^{i\perp}v$ is almost colinear to ${}^{i}w$ for $i = 1, 2, 3$.

V.5.5 Implementation Issues.

Complexity Analysis.

Once the J^T is available, the cost of a PD+gravity controller (Eq. V.38) is only 27 FLOP. Once the accelerations are known in both spaces and the LU decomposition of J^{-1} are available, the simplified model from Eq. V.42 requires only 33 FLOP to be computed, plus 15 FLOP for the PD controller either in joint space or task space. Evaluating the complete model from Eq. V.50 is about three times more expensive (112 FLOP).

For various reasons detailed below, it is not worth the effort trying to parallelize the model evaluation itself.

Load Balancing.

Since the kinematics algorithms could be efficiently parallelized, the central processor is less loaded than the peripherals (see § V.3.5). *It is consequently not advisable to try to share the low burden* associated to the dynamics model and control law on the peripheral processors in order to take advantage of the parallel+pipeline structure of the controller.

Sampling Period and Practical Limitations.

Finally, the 127 FLOP of the complete model based control law are executed in 284 [μs] of T-800 CPU time, giving a traversal time of the complete algorithm of 1604 [μs] on a single T-800 or about 866 [μs] on four, offering a 54% efficiency. Since an overhead of about 100 [μs] is introduced by the communication with the robot's interfaces which are accessed through a Transputer link (see Fig. 25 on page 111), sampling period of 2 [ms] on a single Transputer or 1 [ms] with four T-800 can be achieved.

With the technique of interleaving sampling periods proposed in § IV.7.2, the sampling period can furthermore be reduced to 0.5 [ms] on a control unit with the proposed architecture. However it becomes extremely difficult to design the control software so that no deadlock can occur while dozens of parallel processes exchange two sets of data over a limited number of communication channels. For this reason, the software developed in the framework of this thesis could not reach the required degree of reliability to safely operate the experimental Delta robot with a sampling period shorter than 1 [ms].

Comparison with Former Approaches.

Considering that the complexity of the inverse kinematics algorithms is about 250 FLOP, it can be stated that the complete model of the Delta robot is about 50% more complex than the model of a general 3-DOF *serial arm (275 FLOP [Domb88]). This confirms the results from* § III.7.5, where the complexity of the Newton-Euler algorithm for serial arms and parallel manipulators is compared.

The models derived by Codourey [Codo91a] and Miller [Mill92] require inverse kinematics relations only to map the reference positions onto joint space. Numerical differentiation techniques are applied to obtain velocities and accelerations in both spaces. To allow a comparison with the proposed approach, the cost of these operations (99 FLOP) should be added to the reported complexity of the evaluation of Codourey's simplified model (133 FLOP) and Miller's complete model (206 FLOP) [Mill92].

Therefore, it can be stated that the proposed closed-form models are about 20% more complex than the earlier corresponding models with numerical differentiations. However, unlike previous results that allowed only feedforward dynamics compensation, true computed-torque and resolved-acceleration schemes can be built with the proposed models, which moreover explicitly provide the Jacobian matrix.

V.6 Simulations, Experimental Results and Analysis.

V.6.1 A Simple Test Path.

To illustrate the Delta's possibilities, the robot's motion along a test path is first simulated. The Delta's platform starts from $p_s = \begin{bmatrix}0 & -0.400 & -0.350\end{bmatrix}^T$ [m] and follows a circular path of center $p_c = \begin{bmatrix}0 & 0 & -0.350\end{bmatrix}^T$ in the horizontal Oxy plane until position $p_e = \begin{bmatrix}0 & 0.400\,(0) & -0.350\end{bmatrix}^T$ [m] is reached. A cycloidal velocity profile with maximal accelerations of $120\,[\mathrm{m/s^2}]$ is used (see [Codo91a]).

The positions, velocities and accelerations in operational space are displayed in the left hand side column of Fig. 31. The dotted curves indicate the modulus of the end-effector velocity and acceleration. The inverse kinematics algorithms described in Section V.3 were used with the parameters of the Delta-IA (see § V.2.3) to compute the corresponding positions, velocities and accelerations in joint space, which are shown in the right hand side column of Fig. 31.

The various terms of the inverse dynamics model in the two spaces are represented in Fig. 32. Dotted curves in Fig. 34a represent ${}^q g$, dashed curves represent $J^T \cdot {}^p g$ while solid curves give the sum of both contributions. A similar convention is used in Fig. 32b where ${}^q A \cdot \ddot{q}_c$, $J^T \cdot {}^p A \cdot \ddot{p}_c$ and the sum of both is represented. The Coriolis term $J^T \cdot {}^p C \cdot \dot{p}$ is represented in Fig. 32c. Finally, the total torque obtained through the evaluation of Eq. V.50 is displayed in Fig. 32d with solid curves while dotted curves represent the result of the simplified model from Eq. V.42.

Analysis.

This path of about 1.25[m] is tracked in 0.253[s]. The platform's speed reaches 10[m/s] and, because of the path's curvature, the normal acceleration in the middle of the trajectory is twice the specified maximal tangent acceleration. Consequently, the torques required to accelerate the platform *(dashed curves in Fig. 32b) have about the same magnitude as the torques associated to the acceleration* of the joints (dotted curves in Fig. 32b).

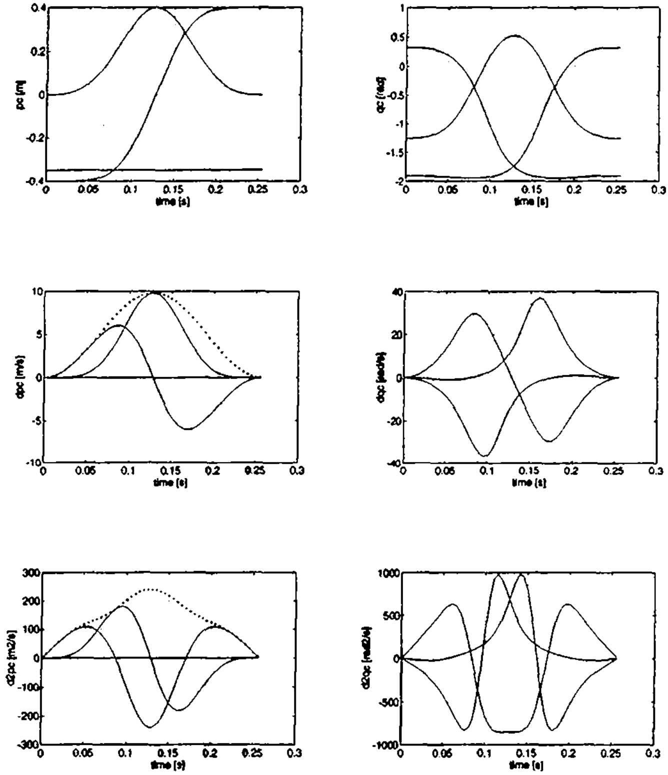


Fig.31 : Simple Test Path: Kinematics.

Even when moving at top speed, the Coriolis terms (Fig. 32c) are one order of magnitude lower than the inertia terms. However, one can remark that they globally reduce the magnitude of the total required torque. This can be seen in Fig. 32d where the results of the simplified model (dotted curves) systematically exceed those from the complete model (solid curves).

A Comment on the Generation of Optimal Paths.

It is interesting to note that the actuators power is very well used on this simple test path: all three torques reach about the same magnitude and, during most of the travel time, at least one is close to its maximal value. It is therefore suspected that this half-circular path is close to the time minimal path between the two diametrically opposed points p_s and p_e.

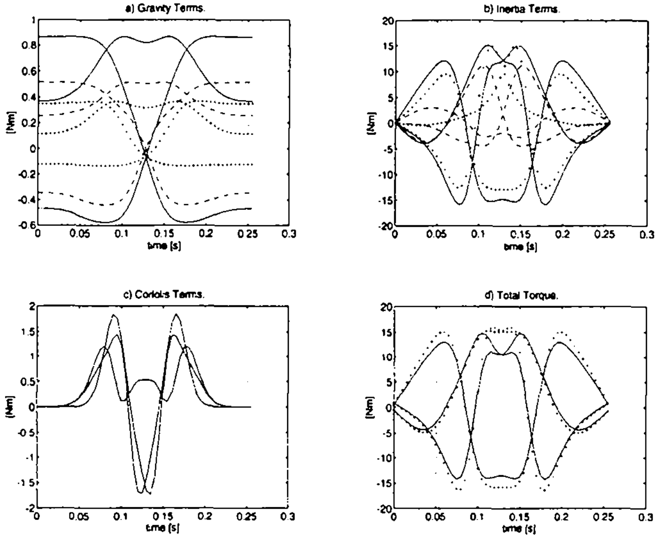


Fig.32 : Simple Test Path: Dynamics.

For fast pick & place manipulators such as the Delta, generating a path that minimizes the travel time between two points is a very challenging problem. Straightforward approaches such as nonlinear programming are known to give the optimal solution, but are computationally very expensive. In this respect, techniques based on successive refinements might be more appropriate. It is suspected that fairly good sub-optimal initial guesses might be obtained by taking the properties of the robot's dynamics into account. In this context, it is clear from the obtained dynamics models that the *eigenvalues of the Jacobian* matrix play a crucial role that should be examined.

In the case of the Delta, a closer look at Fig. 28 can already help the trained user in getting an intuitive idea of the minimal-time paths.

V.6.2 A Simple Pick & Place Task.

A second simulation and analysis is made for the pick & place task experimented by Codourey [Codo91a].

The robot's end effector moves from starting point $p_s = \begin{bmatrix} 0.125 & 0 & -0.350 \end{bmatrix}^T$ [m] to end point $p_e = \begin{bmatrix} -0.125 & 0 & -0.350 \end{bmatrix}^T$ [m] along a half-elliptic path through point $p_t = \begin{bmatrix} 0 & 0 & -0.310 \end{bmatrix}^T$. The path is about 0.292[m] long. Using a cycloidal profile with accelerations up to 80 $[m/s^2]$, it is performed in 0.153[s].

Kinematic values are displayed in Fig. 33 and dynamics in Fig. 34 with the conventions already defined in § V.6.1. Since the trajectory lays in the Oxz plane, the joints number 2 and 3 perform the same motion and the corresponding curves are therefore superimposed.

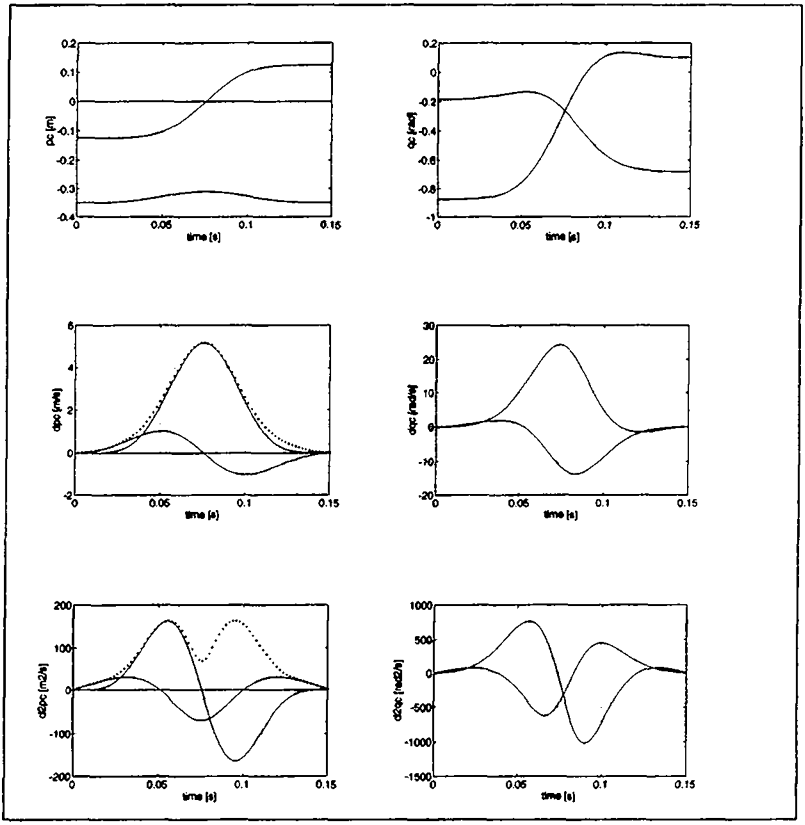


Fig.33 : Pick & Place Task: Kinematics.

Analysis.

The Coriolis effects plotted in Fig. 34c have a low magnitude compared to the inertia terms. Therefore, as expected from the discussion in § V.5.4, the results of the simplified and complete models are very close.

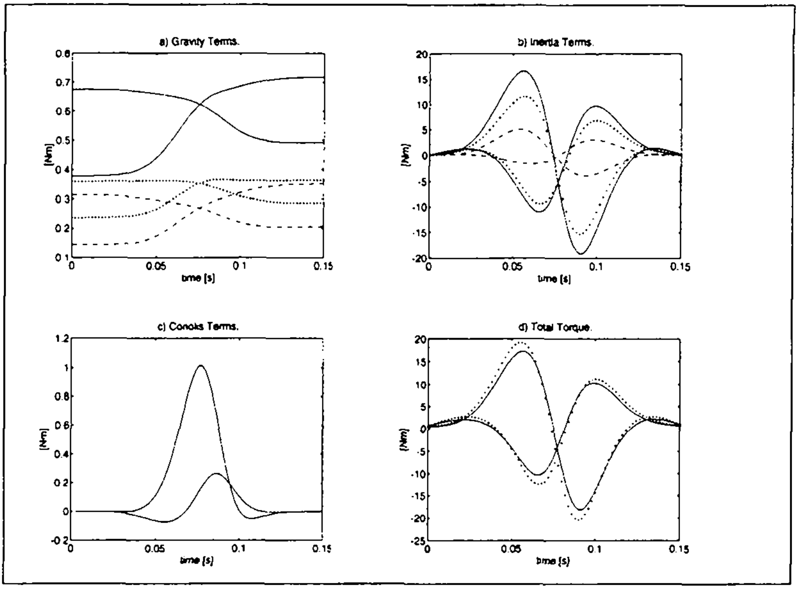


Fig.34 : Pick & Place Task: Dynamics.

Moreover, the centrifugal effects induced by the platform's motion along a curved trajectory also have a limited magnitude (dashed curves in Fig. 34b) because:

- the path is quite short, preventing the platform to reach a really high speed;
- the path is not much curved in its middle part;
- *the path is well centred. Its top lays in a region of the workspace where the Jacobian's eigenvalues are* small (seeFig. 28), reducing the magnitude of the projection onto joint space.

Comment on the Application of the Delta to Pick & Place.

Even if the test path considered here is somewhat ideal, the analysis above remains qualitatively valid for all upwardly curved paths that cross the workspace in its upper part. Since this is typically the case in pick & place tasks, it can be stated that the Delta robot is extremely well suited for this type of applications.

In this context, since the robot's dynamics is dominated by the linear behaviour of the arms, actuators and transmissions (dotted curves in Fig. 34b), it is clear that independent linear controllers on each joint perform satisfactorily. Codourey [Codo91a] obtained a very good tracking performance just by adding a proper feedforward control. As indicated in § IV.3.5, there is no reason to think that a full resolved acceleration control scheme would offer better performance.

On the other hand, these considerations show the vital importance of the choice of the Delta's actuators, sensors and transmissions since the influence of unmodelled dynamics in these elements exceeds the influence of unmodelled dynamics in the structure itself.

V.6.3 Identification of Actuator and Transmission Dynamics.

Actuator Dynamics.

To evaluate the dynamics of the actuators, an identification was performed by exciting the power amplifiers with a pseudo-random binary sequence of torque setpoints and measuring the positions with a 1 KHz sampling frequency. The Bode plot of the identified ARMA model with 4 poles is shown in Fig. 35 in the case of an unloaded actuator.

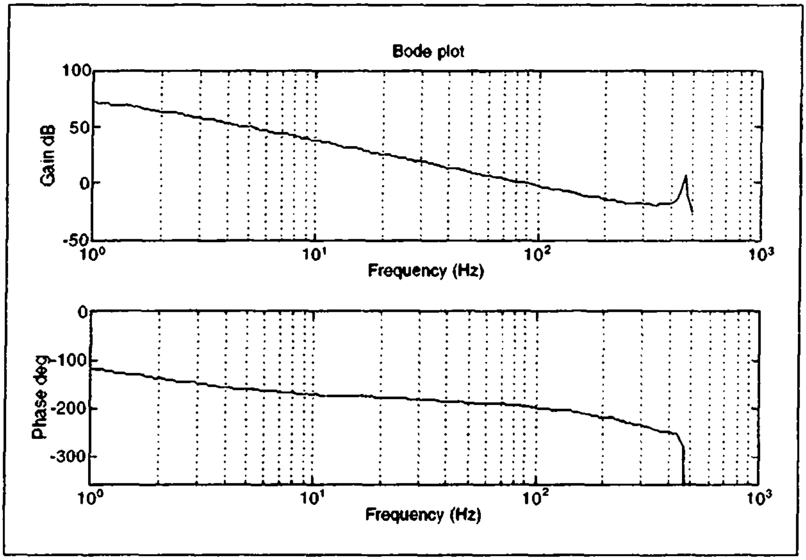


Fig.35 : Unloaded Actuator Frequency Response.

The two dominant poles are the integrator and the viscous friction pole at about 1 Hz (see § IV.2.1). A very badly damped resonance moreover appears. Experiments have shown that the resonance is actually at 550 Hz and folded up to 450 Hz because of the 1 KHz sampling frequency. The cause of this annoying phenomena certainly resides in the current loop implemented in the power amplifiers: the *electrical time constant of the actuators is reported to be equal to 1.86 [ms], which corresponds to a*

538 [Hz] cutoff frequency. Since no guard filter can be introduced before an incremental counter and a high order lowpass digital filter would require a higher sampling frequency, this resonance can not be cancelled and strongly limits the performances of the robot.

Transmission and Arm Dynamics.

When the arm is driven through the reduction, the mechanical resonance appears (see § IV.2.1). Since the sensors are mounted on the actuators, two zeros also appear in the model (see § IV.2.1). The Bode plot of the identified ARMA model with 6 poles and 2 zeros corresponding to this case is given in Fig. 36.

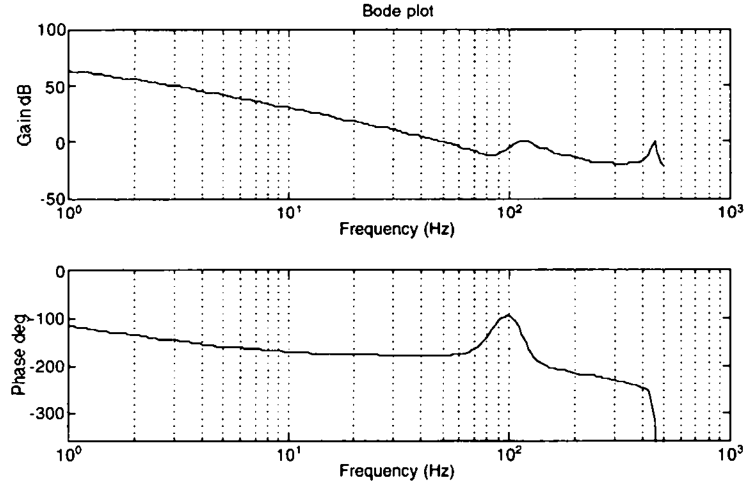


Fig.36 : Actuator + Transmission + Arm Frequency Response.

The identified inertia is

$$j_a = 0.0131\ [\mathrm{Kg.m^2}], \tag{V.52}$$

and the resonance frequency is about 125 [Hz].

From these values, the elasticity constant of the transmission can be evaluated to $k_e \cong 8080[\mathrm{N.m/rad}]$, which is 38% less than the value obtained from the belt's specifications. The belt's stiffness can be raised by increasing its tension, but experiments have shown that the resonance's damping then becomes lower, counterbalancing the advantage of a higher natural frequency.

Nominal Natural Frequency of a Single Chain.

Neglecting the arm's flexibility and considering each chain in the configuration where the forearm is orthogonal to the arm, the joint's inertia becomes $j_a = 0.0253$ [Kg.m^2] and the associated natural frequency of the open chain can be evaluated to about 88 Hz. Note that the actuator's inertia at joint level is only $j_m = 0.0065$ [Kg.m^2], indicating that a more powerful motor with an higher inertia or a larger reduction ratio would give a better balanced oscillator.

V.6.4 Applicability of the Proposed Control Scheme.

The performance of the Delta prototype available for the experimental part of this work is limited by the very poor behaviour of the actuators power amplifiers and transmissions. The velocity step response of a single actuator is given in Fig. 37, where the presence of an undamped oscillation is clearly visible.

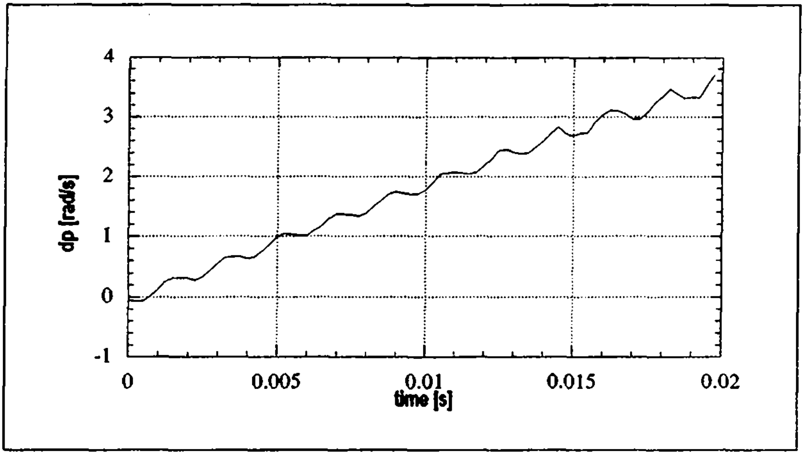


Fig.37 : Actuator Velocity Step Response.

In this context, it is useless to design advanced control laws that take the manipulator dynamics into account. In practice, preliminary experiments have shown that the proposed control schemes perform worse than independent PD control on each joint. More successful implementations of model-based control laws have been reported on newer, direct-drive Delta prototypes [Codo91a].

On the other hand, commercial versions of the Delta reach comparable performances with very high reduction ratios, where the nonlinear part of the dynamics can be neglected or compensated through proper feedforward control. *Learning* techniques [Tso 93] are likely to be very useful in practice since the Delta is most often applied to repetitive tasks.

V.7 Conclusions.

A systematic approach is used to model the Delta parallel robot. The constants that parametrize the manipulator's geometry and dynamics are enumerated. Their numerical values for the prototype Delta manipulator available at IA-EPFL are given.

Then, through extensive use of unit vectors along the bodies building the Delta structure, inverse kinematics models for position, speed and acceleration mappings are proposed that result in very efficient, parallelizable algorithms. A first analysis of the robot's performance is proposed based on a numerical study of the Jacobian matrix. A coherent definition of the robot's usable workspace is derived.

The inverse dynamics model in the two spaces of the Delta is then developed in three steps: the gravity compensator is obtained, then a simplified model with neglected inertia momentum of the forearms is established to finally obtain a complete model in a very handy and compact formulation.

Simulation results are reported that allow some important remarks on the dynamics of the Delta manipulator and its influence on applications. Finally, experimental identification results of the actuator's dynamics are displayed. Because of the very poor behaviour of the power amplifiers and transmissions, the available Delta prototype could not be operated correctly with the proposed control laws.

Chapter VI

Conclusions.

The five sentences in the thesis from Section I.1 propose a general framework for the model-based control of fast parallel manipulators. A series of original contributions and clearer formulation of known results are then developed in this report to back up these five sentences.

A wide class of parallel robots can be modelled as two partial tree structures connected through simple links.

Parallel manipulators are not general mechanical structures: as any other robot, they are specifically designed to perform a given task with their end-effector while their structure transmit forces and torques from the actuators to the end-effector.

Based on considerations about the mobility of a structure with closed kinematic chains, a wide class of parallel manipulators is defined. Most of the existing parallel manipulators such as the so-called Stewart platform and its variations, the Pollard, Hexa, Speed-R-Man and Hexa belong to this class.

A formalism inspired from the Denavit-Hartenberg parametrization is proposed, that allows a systematical parametrization of the geometry of such manipulators.

The corresponding dynamics model is called "in the two spaces" since it uses the robot's state simultaneously in joint space and in operational space.

Parallel manipulators are constrained systems. They are many different ways to parametrize their dynamics and to define the corresponding constraints. For the manipulators belonging to the defined class, the kinetic and potential energy of the constrained system can be systematically and simply obtained using a vector dependent generalized coordinates that contains both the variables associated to the actuated joints and the variables describing the end-effector's position in operational space. In this case, the constraints can simply be represented by the inverse kinematics equations relating the joint positions to the end-effector's position.

Using this formalism, it is shown that the standard "Lagrange with multipliers" approach leads to a very handy and compact formulation of the manipulator's inverse dynamics model called "in the two spaces" where the manipulator's Jacobian matrix explicitly appears.

An algorithm that allow real-time implementation of the proposed inverse dynamics model is proposed. It is based on a based the Newton-Euler method derived by Luh, Walker and Paul's for serial arms. The analysis shows that the evaluation of the dynamics of a parallel manipulator is much more expensive than for a serial arm.

This formulation makes it possible to find passive mappings in the dynamics of parallel manipulators.

The analysis of the inverse dynamics model in the two spaces allows to show the existence of a passive mapping from the forces applied on the end-effector and its velocity. It is shown that a passive mapping from the torques on the actuated joints to the joints velocity exist only locally, in sub-spaces where the Jacobian matrix remains bounded. Therefore, all the control laws developed for serial arms can be applied to parallel robots as long as they are operated far from "inverse singularities" that do not exist in serial arms.

A robust, model-based control law can therefore be designed in operational space to achieve high-performance trajectory tracking.

Among other control strategies, model-based schemes have shown their superiority in many practical applications. The inverse dynamics model in the two spaces makes it easy to design such schemes with a feedback loop either in joint space (computed torque) or in operational space (resolved acceleration). It is shown that, among other advantages, this latter approach requires less computing power than its equivalent in joint space.

The intrinsic parallel-pipeline structure of the corresponding algorithms can be exploited on a *multiprocessor control unit*.

Many algorithms involved in a model-based control scheme of a parallel robot are shown to have an intrinsic parallel structure that allow to efficiently implement them on a multiprocessor control unit. However, some parts of the algorithms can not be parallelized and should be implemented on a single processor. Since about half of the computing time is spent in parallelized parts and the other half in sequential parts, it is possible to halve the sampling time by interleaving the computations related to two consecutive sampling intervals. A multiprocessor control unit with a star topology is required to implement this two-stages parallel-pipeline structure.

Application to the Delta Robot.

A prototype of the Delta robot is taken as an application example for whole proposed approach.

First, inverse kinematics models are developed systematically using unit vectors attached to the arms and forearms of the Delta. The resulting formulation allows analysis of the properties of the structure over the robot's workspace. It is shown that the Delta offers a high stiffness and low inertia only in a limited part of its workspace. A "usable workspace" where the robot's performances satisfy given requirements should therefore be carefully defined. A very simple and conservative solution is proposed.

Then, the Delta's inverse dynamics model in the two spaces is obtained in three steps. The gravity compensation terms are derived first, then a simplified model equivalent to the one proposed by Codourey is obtained. It neglects the inertia momentum of the robot's forearms. This hypothesis is then removed in a complete model, which is equivalent to Miller's but explicitly introduces closed-form kinematics relations instead of using numerical techniques.

The complexity of the various proposed algorithms is analysed and computing time obtained on the experimental control unit with 4 Transputers are given. A sampling frequency of 1[KHz] can be reached for the complete model-based control law.

Finally, simulation results lead to interesting remarks about the robot's performance and application field. Some experimental results obtained when identifying the dynamics of the Delta prototype are given. They show a very bad performance of the available power amplifiers that prevented satisfactory experimental results in closed loop to be obtained.

Appendix I

Notations.

Scalars, Vectors, Matrices, Sets and Vector Spaces.

- Capital letters except O denote matrices.
- Lower case symbols except i, j, n, m, s, t, z, π indicate vectors.
- i, j, n, m, s, t, z, π denote integer, real or complex scalar numbers.
- Sets and vector spaces are specified by cursive capitals, like $\mathcal{A}$.

Functions, Expressions, Sets.

- Most of the vectors and matrices encountered in this document are also functions of some parameters. A parenthesized list of parameters $(\dots)$ is specified in the definition of each symbol. However for the sake of clarity, the parameters are often omitted in the developments.
- In expressions, brackets $[\dots]$ are used to enclose vector or matrix terms while parenthesis denote scalar terms.
- Sets are defined using braces $\{\dots\}$.

Matrix Operators, Subscripts and Superscripts.

- Subscript c denotes a reference signal.
- Subscript b denotes a feedback signal.
- Subscript f denotes a feedforward signal.
- i, j or numbered subscripts are used to specify scalar elements of vectors, column vectors in matrices.
- Subscripts on scalars are used as specifiers : e_P denotes the potentialenergy, e_K the kinetic energy.
- Pre-superscripts are used to denote blocks of a matrix that correspond to a projection of the matrix onto a subspace of smaller dimension e.g. ${}^{q}A$ and ${}^{p}A$.
- Matrix transposition is denoted by a T superscript like in J^T.

- Combined matrix inversion and transposition uses the convention $J^{T-1} = J^{-1T} = J^{-T}$.

Diacritical Marks.

- Time derivatives are denoted by doted symbols like $\ddot{p}$ or $\dot{A}$.
- Hats are used to denote estimated signals, e.g. $\hat{M}$.
- Error or deviation signals are specified by tildes : $\tilde{q} = q_c - q$
- A prime is given to functions whose parameters have been converted from joint space to operational space : $C'(p, \dot{p}) = C(f(q), J \cdot \dot{q})$. Primed symbols are formally equal to the corresponding unprimed symbols. This notation is introduced to avoid confusions when parameters are dropped.

Alphabetical List of Symbols with Reference to their Definition.

a	vector of $n + n_e$ integer indexes describing the structure of a tree robot.	§ II.5.3
A	$m \times m$ inertia matrix.	§ III.4.1
${}^{q}A$	$n \times n$ block of A corresponding to joint space.	§ III.6.1
${}^{p}A$	$n \times n$ block of A corresponding to operational space.	§ III.6.1
${}^{o}A$	$n \times n$ inertia matrix in operational space.	§ III.8.2
$\mathcal{A}$	Joint space, subspace of $\mathcal{R}^{n_q}$.	§ II.3.3
b	Vector of the $m - n$ constraints.	§ III.4.1
B	Matrix of the derivatives of the constraints.	§ III.4.1
C	$m \times m$ quadratic forces matrix.	§ III.4.1.
${}^{q}C$	Bloc of C corresponding to joint space.	§ III.6.1
${}^{p}C$	Bloc of C corresponding to operational space.	§ III.6.1
${}^{o}C$	$n \times n$ quadratic forces matrix in operational space.	§ III.8.2
$\mathcal{C}$	Configuration space, subspace of $\mathcal{R}^{n_q}$	§ II.3.3
d_i	Denavit-Hartenberg parameter of the ith joint.	§ II.5.3
D	Pseudo-quadratic forces matrix.	§ III.5.2
e_H	Hamiltonian of the structure.	§ III.8.2
e_K	Kinetic energy of the structure.	§ III.4.1
e_L	Lagrangian of the structure.	§ III.4.1
e_P	Potential energy of the structure.	§ III.4.1
f	Forward kinematics mapping $\mathcal{A} \to \mathcal{O}$.	§ II.8.1
f^{-1}	Inverse kinematics mapping $\mathcal{O} \to \mathcal{A}$.	§ II.8.2

${}^{i,j}f$	Force exerted by the i^{th} body on the j^{th}.	§ III.7.2
F_i	Force exerted on the i^{th} body.	§ III.7.2
${}^{i}g$	Cartesian position of the i^{th} body's center of gravity with respect to the i^{th} frame.	§ III.7.2
g	Vector of m external forces due to gravity.	§ III.4.1
g_a	Position of the center of gravity of the Delta's arm along v.	§ V.2.3
g_b	Position of the ceneter of gravity of the Delta's forearm along w.	§ V.2.3
${}^{q}g$	Bloc of g corresponding to joint space.	§ III.6.1
${}^{p}g$	Bloc of g corresponding to operational space.	§ III.6.1
G_i	Transfer function of the linearized i^{th} joint.	§ IV.2.1
${}^{o}h$	Gravity forces vector in operational space.	§ III.8.2
h	Pseudo-gravity vector.	§ III.5.2
${}^{i}I$	Inertia matrix of the i^{th} body with respect to the origin of the i^{th} frame.	§ III.7.2
j_a	Inertia of one of the Delta's arms at the joint level, including actuator and transmission.	§ V.2.3
j_b	*Inertia of one of the Delta's forearms about its center of gravity.*	§ V.2.3
j_m	Inertia of the actuator.	§ IV.2.1
j_l	Inertia of the load.	§ IV.2.1
J	Jacobian $n \times n$ matrix.	§ II.9.1
k_e	Elasticity constant of a transmission.	§ III.3.1
k_t	Torque constant of the actuator.	§ IV.2.1
K_p	Diagonal matrix of proportional gains.	§ IV.2.1
K_d	Diagonal matrix of derivative gains.	§ IV.2.1
l_a	Length of the Delta's arms.	§ V.2.2
l_b	Length of the Delta's forearms.	§ V.2.2
l_e	Half the distance between each pair of rods ofthe Delta's forearms.	§ V.2.2
l_i	Length of the i^{th} link.	§ II.7.5
L	Stiffness matrix.	§ III.3.1
m	Number of generalized coordinates w.	§ III.4.1
m_a	Mass of one of the Delta's arms.	§ V.2.3
m_b	Mass of one of the Delta's forearms.	§ V.2.3
m_c	Mass off the Delta's platform and carried load.	§ V.2.3

m_i	Mass of the ith body.	§ III.7.2
M	Pseudo-inertia matrix.	§ III.5.2
n	Mobility of an holonomic non redundant robot.	§ II.3.6
n_a	Number of assembly modes.	§ II.4.1
n_b	Number of rigid bodies in the robot's structure.	§ II.2.1
n_e	Number of end-effectors in the robot's structure.	§ II.5.2
n_j	Number of joints in the robot's structure.	§ II.2.1
n_i	Mobility of the i^{th} joint.	§ II.2.1
n_q	Mobility of the robot's structure.	§ II.3.1
n_p	Mobility of the robot's end-effector.	§ II.3.4
n_r	Reduction ratio of the transmission.	§ IV.2.1
n_s	Number of sub-chains in a parallel manipulator.	§ II.6.2
${}^{b}n$	Number of actuated joints in the base tree.	§ II.7.4
${}^{u}n$	Number of actuated joints in the upper tree.	§ II.7.4
${}^{l}n_i$	Number of actuated joints in the i^{th} link.	§ II.7.4
n_φ	Number of φ parameters.	§ III.4.2
N_i	*Torque exerted on the i^{th} body.*	§ III.7.2
O	operational space, subspace of $\mathcal{R}^{n_p}$.	§ II.3.4
${}^{i,j}p$	Cartesian position of the i^{th} frame's origin with respect to the j^{th} frame.	§ II.5.3
p	*Position vector of the end-effector in operational space $p \in O$.*	§ II.3.4
${}^{b}P_i$	i^{th} connection point on the base-tree.	§ II.7.4
${}^{u}P_i$	i^{th} connection point on the upper-tree.	§ II.7.4
q	*Position vector in joint space $q \in \mathcal{A}$.*	§ II.3.3
q_{min} and q_{max}	Bounds on the Delta's joints positions.	§ V.4.2
r_a	Radius of the Delta's base.	§ V.2.2
r_a	*Radius of the Delta's platform.*	§ V.2.2
r_r	Characteristic radius of the Delta.	§ V.2.2
r_i	Denavit-Hartenberg parameter of the i^{th} joint.	§ II.5.3
${}^{i}R$	*Rotation matrix of the i^{th} auxiliary frameof the Delta robot.*	§ V.3.1
${}^{i,j}R$	3×3 rotation matrix of the direction cosines of the i^{th} frame's base vectors with respect to the previous j^{th} frame.	§ II.5.3

s	Laplace-transform variable (complex),	
	or curviliear abscissa along the trajectory (scalar).	§ II.10.1
t	Time domain variable (scalar).	§ II.10.1
t_0	Initial time.	§ II.10.1
t_f	Final time.	§ II.10.1
$t_{m,i}$	Dominant time constant of the i^{th} actuator.	§ IV.2.1
${}^{i}T$	Homogeneous (4×4) matrix defining the i^{th} joint's transformation with respect to the a_i^{th}.	§ II.5.3
u	Control command. Input of the robot's power amplifiers.	
${}^{i}v$	Unit vector along the i^{th} link of a general manipulator.	§ III.7.4
	Unit vector along the i^{th} arm of the Delta robot.	§ V.3.1
V	Base matrix formed by base vectors along the Delta's arms.	§ V.3.1
w	Vector of the m generalized coordinates	§ III.4.1
	or natural coordinates.	§ II.6.5
${}^{i}w$	Unit vector along the i^{th} forearm of the Delta robot.	§ V.3.1
W	Base matrix formed by base vectors along the Delta's forearms.	§ V.3.1
${}^{i}x$	Unit vector attached to the i^{th} joint.	§ II.5.3
X	Auxiliary diagonal matrix useful in the Delta's modeling process.	§ V.3.3
Y	$(n \times n_\varphi)$ matrix of the nonlinear functionsin dynamics	§ III.4.2
z	z-transform variable.	§ II.10.3
${}^{i}z$	Unit vector attached to the i^{th} joint.	§ II.5.3
α_i	Denavit-Hartenberg parameter of the i^{th} joint.	§ II.5.3
$\bar{\beta}$, $\underline{\beta}$	Upper and lower bounds.	§ V.4.2
β	Kinetic friction coefficient in a transmissiont.	§ IV.2.1
β_l	Kinetic friction coefficient at thge load level.	§ IV.2.1
β_m	Kinetic friction coefficient at the actuator level.	§ IV.2.1
γ_i	Extended Denavit-Hartenberg parameter of the i^{th} joint.	§ II.5.3
Γ	$m \times m \times m$ tensor of the inertia matrix' Christoffel symbols.	§ III.4.1
ζ_i	Extended Denavit-Hartenberg parameter of the i^{th} joint.	§ II.5.3
ς	Vector of m generalized forces/torques applied on the structure.	§ III.4.1
θ_i	Denavit-Hartenberg parameter of the i^{th} joint.	§ II.5.3

ι	Curvilinear abscissa along the trajectory.	§ II.10.2
λ	Vector of the $m - n$ Lagrange multipliers.	§ III.4.1
σ_i	i^{th} joint's prismatic/revolute specifier.	§ II.5.3
${}^{i,j}\eta$	Torque exerted by the i^{th} body on the j^{th}.	§ III.7.2
τ	Vector of torques/forces applied on the actuated joints. $\tau \in \mathcal{A}$	Section III.3
υ	Vector of torques/forces applied on the end-effector. $\upsilon \in \mathcal{O}$	Section III.3
φ	Vector of the n_φ parameters.	§ III.4.2
${}^{i}\omega$	Vector of rotation about the center of gravity of link i.	§ III.7.4
${}^{b}\Omega_i$	Vector of rotation about ${}^{b}P_i$.	§ III.7.4
${}^{u}\Omega_i$	Vector of rotation about ${}^{u}P_i$.	§ III.7.4
${}^{i,j}\omega$	Vector of the i^{th} frame's rotation velocity with respect to the j^{th} frame.	Section III.7

Appendix II

List of Figures.

Fig.1 : A Complete Robotized Plant. 19
Fig.2 : The Delta Robot. 22
Fig.3 : Joint Space Control or Operational Space Control 24
Fig.4 : Three Degrees of Freedom Serial Arm. 31
Fig.5 : "Three plus Two" DOFs Tree–Robot. 32
Fig.6 : 3–DOF Robot with a "5–bars Mechanism". 34
Fig.7 : Three degrees of freedom parallel robot. 35
Fig.8 : A Fancy 6 DOFs Parallel Manipulator. 38
Fig.9 : Conversion of Reference Signals. 51
Fig.10 : Link with Colinear and Orthogonal Decomposition of Velocities. 69
Fig.11 : Actuated Joint with Elastic Transmission. 80
Fig.12 : Typical Open Loop Frequency Responses of a Serial Arm Joint.
(Elbow joint of a Puma 560 [Tarn93])82
Fig.13 : Typical Open-Loop Frequency Responses of a Serial Arm Joint with PD control. 84
Fig.14 : Typical Open-Loop Frequency Responses of a Serial Arm Joint with PID control. 85
Fig.15 : Feedback Linearization in Joint Space. 87
Fig.16 : Computed-Torque Control Scheme. 88
Fig.17 : Feedback Linearization in Operational Space. 91
Fig.18 : Computed Torque Scheme for a Parallel Manipulator. 103
Fig.19 : Resolved-Acceleration Scheme for a Parallel Manipulator. 103
Fig.20 : Coarse Grain Structure of the Resolved Acceleration Scheme for Parallel Robots. . . . 106
Fig.21 : Fine Grain Parallel Structure. 107
Fig.22 : Processor Usage over a Single Sampling Interval. 107
Fig.23 : Obtaining a Balanced 50% CPU Usage. 108
Fig.24 : Interleaving Sampling Intervals to obtain 100% CPU Usage
in a Two-Stages Pipelined Control Unit.109
Fig.25 : Structure of the Experimental Control Unit. 111
Fig.26 : The Delta Robot. 114
Fig.27 : Geometry of the Delta Robot. 117
Fig.28 : Jacobian's Determinant on Seven Equidistant Horizontal Slices of the Operational Space.126

Fig.29 : Minimum eigenvalue of as a function of the bounds of 128
Fig.30 : Usable Workspace of the Delta 740. 128
Fig.31 : Simple Test Path: Kinematics. 136
Fig.32 : Simple Test Path: Dynamics. 137
Fig.33 : Pick & Place Task: Kinematics. 138
Fig.34 : Pick & Place Task: Dynamics. 139
Fig.35 : Unloaded Actuator Frequency Response. 140
Fig.36 : Actuator + Transmission + Arm Frequency Response. 141
Fig.37 : Actuator Velocity Step Response. 142

Appendix III

List of Tables.

Table 1 :Dimensions of Typical Parallel Manipulators.. 39
Table 2 :Table of Equivalent Abstract Links. 40
Table 3 :Null Length Equivalent Abstract Links.. 40
Table 4 :Comparative Kinematics: Serial versus Parallel Structures. 52
Table 5 :Reported Experimental Setups for Model-Based Control of Serial Robots. 93
Table 6 :Reported Experimental Setups for Model-Based Control of Parallel Robots. 101
Table 7 :Parametrization of the Delta Manipulator Base Tree. 115
Table 8 :Parametrization of the Delta Manipulator Upper Tree. 116

Appendix IV

References.

[Abda91] C. Abdallah, D. Dawson, P. Dorato and M. Jamshidi, "Survey of Robust Control for Rigid Robots", 1991, IEEE Control Systems, Vol. Feb, pp. 24-29

[Adli91] M-A Adli, K Nagai, K Miyata and H. Hanafusa, "Apparent Structural Stiffness of Closed Mechanisms Under The Effect of Internal Forces During Dynamic Motion.", 1991, Proceedings of Proceedings IROS '91. IEEE/RSJ International Workshop on Intelligent Robots and Systems '91. Intelligence for Mechanical Systems, Osaka, Japan, Vol. 2, No. 3, pp. 773-8

[Adli90] M-A Adli, K Nagai, K Miyata, H Hanafusa and O. Ed. by Kaynak, "Effect of Internal Forces on the Dynamics of the Parallel Manipulators.", 1990, Proceedings of Intelligent Motion Control. Proceedings of the IEEE International Workshop, Istanbul, Turkey, Vol. 2, No. 2, pp. 547-50

[Agra90] Sunil-K. Agrawal, "Equations of Motion of in-Parallel Manipulator Systems.", American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC v 26. Publ by ASME, New York, NY, USA, pp. 213-218

[Ait-92] M. Ait-Ahmed and M. Renaud, "New Geometric Parameters for the Modelling of Simple or Closed-Chain Mechanisms", 1992, Proceedings of IFAC Workshop on Motion Control for Intelligent Automation, Perugia, Italy, pp.29-32

[Ait93] M. Ait-Ahmed, "Contribution à la modélisation géométrique et dynamique des robots parallèles", 1993, Thèse de doctorat, Université Paul Sabatier, Toulouse 2

[An89] C.H. An, C.G. Atkeson, J.D. Griffith and J.M. Hollerbach, "Experimental Evaluation of Feedforward and Computed Torque Control", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, pp. 368

[Ande89] R.J. Anderson, "Passive Computed Torque Algorithms for Robots", 1989, Proceedings of IEEE Conf. Dec. & Control, Tampa, Fl, USA, pp.1638-1644

[Ange88] J. Angeles and S.-K. Lee, "The Formulation of Dynamical Equations of Holonomic Mechanical Systems Using a Natural Orhogonal Complement", 1988, ASME J. of Applied Mechanics, Vol. 55, No. 1

[Asad86] H. Asada and J.J. Slotine, "Robot Analysis and Control", Wiley, New York, 1986

[Aubi91] A. Aubin, C. Canudas de Wit and H. Sidaoui, "Dynamic Model Simplification of Industrial Robot Manipulators", 1991, Proceedings of SYROCO, Vienna, Austria, Vol. 1, No. 2, pp. 9-14

[Baha91] I.B. Baharin and R.J. Green, "Computationally-Effective Recursive Lagrangian Formulation of Manipulation Dynamics", 1991, Int. Journal of Control, Vol. 54, No. 1, pp. 195-214

[Baya88] David S. Bayard and John T. Wen, "New Class of Control Laws for Robotic Manipulators. Part 2 : Adaptive Case", 1988, Int. Journal of Control, Vol. 47, No. 5, pp. 1387-1406

[Bala88] C.A. Balafoutis, R.V. Patel and J. Angeles, "*A Comparative Study of Newton-Euler, Euler-Lagrange and Kane's Formulation for Robot Manipulator Dynamics*", *1988*, Proceedings of 2nd International Symposiumon Robotics and Manufacturing, Albuquerque, NM, USA, pp.13-20

[Beck88] N. Becker and W. M. Grimm, "On L2 and H∞ Stability Approaches for the Robust Control of Robot Manipulators", 1988, IEEE Transactions on Automatic Control, Vol. 1

[Bobr88] *James E. Bobrow, "Optimal Robot Path Planning Using the Minimum-Time Criterion"*, 1988, IEEE Transactions on Robotics and Automation, Vol. 4, No. 4, pp. 443

[Bokh88] Shahid H. Bokhari, "Partitioning Problems in Parallel, Pipelined and Distributed Computing", *1988, IEEE Transactions on Computers*, Vol. 37, No. 1, pp. 48-57

[Brau90] Hans Brauchli and René Weber, "Dynamical Equations in Natural Coordinates", 1990, Proceedings of 2nd World Congress on Computational Mechanics, Stuttgart

[Brau91] Hans Brauchli, "Mass-Orthogonal Formulation of Motion for Multibody Systems", 1991

[Cair87] R. Cairoli, "Algèbre Linéaire I et II", Presses Polytechniques Romandes, Lausanne, 1987

[Canu91] C. Canudas_de_Wit and J.-J. E. Slotine, "Sliding Observers for Robot Manipulator", 1991, Automatica, Vol. 27, No. 5, pp. 859-864

[Canu92] C. Canudas_de_Wit, N. Fixot and K.J. Astrom, "Trajectory Tracking in Robot Manipulators via Nonlinear Estimated State Feedback", 1992, IEEE Transactions on Robotics and Automation, Vol. 8, No. 1, pp. 138-144

[Chan89] Po Rong Chang and C.S. George Lee, "Residue Arithmetic VLSI Array Architecture for Manipulator Pseudo-Inverse Jacobian Computation", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 5, pp. 569-582

[Chen88] Chun Lung Chen, C.S. George Lee and Edwin S.H. Hou, "Efficient Scheduling Algorithms for Robot Inverse Dynamics Computation on a Multiprocessor System", 1988, TSMC, Vol. 18, No. 5, pp. 729-743

[Chen91] Fan Tien Cheng and David E. Orin, "Efficient Formulation of the Force-Distribution Equations for Simple Closed-Chain Robotic Mechanisms.", 1991, IEEE-Transactions-on-Systems-Man-and-Cybernetics (USA), Vol. 21, No. 1, pp. 25-32

[Chou92] Jack C. Chou, "Quaternion Kinematik and Dynamic Differential Equations", 1992, IEEE Transactions on Robotics and Automation, Vol. 8, No. 1, pp. 53-64

[Clav85] R. Clavel, "Dispositif pour le déplacement et le positionnement d'un élément dans l'espace", Swiss Patent nr 672089 A5, December 1985

[Clav88] R. Clavel, "Nouvelle structure de manipulateur pour la robotique légère", 1988, APPII.

[Clav91] Reymond Clavel, "Conception d'un robot parallele rapide à 4 degrés de liberté", 1991, Doctorat ès sciences techniques EPFL - Département de Microtechnique

[Codo91a] Alain Codourey, "Contribution à la commande des robots rapide et précis", 1991, Doctorat ès sciences techniques EPFL - Département de Microtechnique

[Codo91b] Alain Codourey, R. Clavel and C.W. Burkhardt, "Control Algorithm and Controller for the Direct Drive Delta Robot", 1991, Proceedings of SYROCO, Vienna, Austria, Vol. 1, No. 2, pp. 169-175

[Colb91] R. Colbaugh and K. Glass, "On Controlling Robots with Redundancy.", 1991, Proceedings of Proceedings of the 1991 American Control Conference, Boston, MA, USA, Vol. 2, No. 3, pp. 2059-61

[Cox89] D.J. Cox and D. Tesar, "The Dynamics Model of a Three-Degree-of-Freedom Parallel Robotic Shoulder Module", 1989, Proceedings of the 4th ICAR, Columbus, Ohio, USA, pp. 475-487

[Crai86] J.J. Craig, PI. Hsu and S. Sastry, "Adaptive Control of Mechanical Manipulators", 1986, Proceedings of IEEE Conf. on Robotics and Automation, San Francisco, California, USA, Vol. pp.

[Crai88] John J. Craig, "Adaptive Control of Mechanical Manipulators", Addison-Wesley, Reading, Massachusetts, 1988

[Dahl90] Ola Dahl and Lars Nielsen, "Torque-Limited Path Following by On-Line Trajectory Time Scaling", 1990, IEEE Transactions on Robotics and Automation, Vol. 6, No. 5, pp. 554-561

[Dahl92] Ola Dahl, "Path Constrained Robot Control", Sweden, KF-Sigma, 1992

[Dauc88a] Pierre Dauchez, Alain Fournier and René Jourdan, "Hybrid Control of a Two Arm Robot : The Relative Motion Problem", 1988, Proceedings of 2nd International Symposiumon Robotics and Manufacturing, Albuquerque, NM, USA, pp.361-370

[Dauc88b] Pierre Dauchez and Masaru Uchiyama, "Coordonnées de l'espace de travail pour la commande mixte position/force d'un robot à deux bras", 1988, RAIRO APII, Vol. 22, No. 6, pp. 569-595

[Deso75] C. Desoer and M. Vidyasagar, "Feedback Systems : Input-Output Properties", Academic Press, New-York, 1975

[Do88] W.Q.D. Do and D.C.Il Yang, "Inverse Dynamic Analysis and Simulation of a Platform Type of Robot", 1988, J. of Robotics Systems, Vol. 5, No. 3, pp. 209-227

[Domb88] Etienne Dombre and Wisama Khalil, "Modélisation et commande des robots", Hermès, Paris, 1988

[Eppi93] Steven D. Eppinger and Warren P. Seering, "Three Dynamic Problems in Robot Force Control", 1993, IEEE Transactions on Robotics and Automation, Vol. 8, No. 6, pp. 751-758

[Faes88] Hanspeter Faessler, "Robot Control in Cartesian Space with Adaptive Nonlinear Dynamics", 1988, Proceedings of IUTAM/IFAC Symposium on the dynamics of controlled mechanical systems, Zürich

[Feat87] Roy Featherstone, "Robot Dynamics Algorithms", Kluwer, Boston, 1987

[Fija88] Amir Fijany, "Algorithmes et architectures parallèles en robotique", 1988, PHD Paris Sud

[Fija89] Amir Fijany and Antal K. Bejczy, "A Class of Parallel Algorithms for Computation of the Manipulator Inertia Matrix", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 5 Special Issue on Algorithms and Architecture, pp. 600-615

[Freu82] E. Freund, "Fast Nonlinear Control with Arbitrary Pole Placement for Industrial Robots and Manipulators", 1982, Int. Journal of Robotics Research, Vol. 1, pp. 65-78

[Fuji91] K. Fujimoto et al.,"Derivation and Analysis of Equations of Motion for a 6 DOF Direct Drive Wrist Joint", 1991, Proceedings of IROS 91', Osaka, Japan, pp. 779-784

[Geng92] Z. Geng and L.S. Haynes, "On the Dynamic Model and Kinematic Analysis of a Class of Stewart Platforms"", 1992, Robotics and Autonomous Systems, No. 9, pp. 237-254

[Goss91] Clément M. Gosselin and Jaouad Sefrioui, "Polynomial Solutions for the Direct Kinematic Problem of Planar Three-Degree-of-Freedom Parallel Manipulators", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1124-1129

[Goug62] V.E. Gough and S.G: Whitehall, "Universal tyre test machine", 1962, Proceedings of 9th International Automobile Technical Congress, FISITA, IMechE London, pp.117-137

[Grah89] James H. Graham, "Special Computer Architectures for Robotics : Tutorial and Survey", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 5, Special Issue on Algorithms and Architecture, pp. 543-554

[Gugl91] Ph. Guglielmetti, "Jacobien du Delta", 1991, Rapport Interne 1991-3, Institut d'Automatique IA-EPFL

[Gugl92] Ph. Guglielmetti, "Task Space Control of the Delta Parallel Robot", 1992, Proceedings of the IEEE Workshop on Motion Control for Intelligent Automation, Perugia, Italy.

[Hans88] Herbert Hanselmann, "Low-Resolution Implementation of High-resolution Position Control", 1988, IEEE Transactions on Automatic Control, Vol. 11, pp. 1074

[Hart64] R.S. Hartenberg and J. Denavit, "Kinematic Sythesis of Linkages", McGraw-Hill, New-York, 1964

[Heli90] A.L. Helinski, "Dynamic and Kinematic Study of a Stewart Platform using Newton-Euler Techniques", Tank Automotive Command, Research Report 13479, Jan. 1990

[Herv91] J.-M. Hervé and F. Sparacino, "Structural Synthesis of Parallel Robots Generating Spatial Translation", Proceedings of ICAR, 1991, Pisa, Italy, pp.808-813

[Hsia91] T.C. Hsia, K.W. Current, Z. Mao, W.S Chu, J. Liu, G.Z. Lu and W.H. Han, "A Proposed New VLSI Architecture for Real-Time Robot Manipulator Control", 1991, Int. Journal of Robotics and Automation, Vol. 6, No. 4, pp. 169-178

[Ish-89] Jehuda Ish-Shalom and Peter Kazanzides, "SPARTA: Multiple Signal Processors for High-Performance Robot Control", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 5, pp. 628-640

[Isid83] Alberto Isidori, "Nonlinear Control Systems : an Introduction", Springer-Verlag, Berlin, 1983

[Jalo87] J. Garcia de Jalon, J. Unda, A. Avello and J.M. Jimenez, "Dynamic Analysis of Three-Dimensional Mechanisms in "Natural" Coordinates", 1987, Transactions of the ASME

[Java88] Masoud Amin Javaheri and David E. Orin, "A Systolic Architecture for the Computation of the Manipulator Inertia Matrix", 1988, TSMC, Vol. 18, No. 6, pp. 939-951

[Ji93] Z. Ji, "Study of the Effect of Leg Inertia in Stewart Plaform", 1993, Proceedings of the IEEE Int. Conf. on Robotics and Automation, Atlanta, USA, pp. 121-126

[Kane85] T.R. Kane and D.A. Levinson, "Dynamics : Theory and Applications", McGraw-Hill, New-York, 1985

[Kari92] *J.P. Karidis, G. McVicker, J.P. Pawletko, L.C. Zai, M. Goldowsky, R.E. Brown and* R.R. Comulada, "The Hummingbird Micropositionner – Providing Three-Axis Motion At 50 G's With Low Reactions", 1992, Proceedings of Int. Conf. on Robotics and Automation, Nice, France, pp.685-692

[Klei86] Jean-François Kleinfinger and Wisama Khalil, "Dynamic Modeling of Closed Loop Robots", 1986, Proceedings of 16zh ISIR, pp. 401-412

[Khal87] Wisama Khalil and Jean-François Kleinfinger, "Minimum Operations and Minimum Parameters of the Dynamic Models of Tree Structure Robots", 1987, Int. Journal of Robotics and Automation, Vol. RA-3, No. 6, pp. 517-526

[Khat80] Oussama Khatib, "Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles", 1980, Thèse de docteur-ingénieur ENSAE, Toulouse

[Khat85] Oussama Khatib, "The operational Space Formulation in Robot Manipulator Control", 1985, Proceedings of 15th ISIR, Tokyopp.165-172

[Khat87] Oussama Khatib, "A Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation", 1987, Int. Journal of Robotics and Automation, Vol. 3, No. 1

[Khos89a] Pradeep K. Khosla, "Categorisation of Parameters in the Dynamic Robot Model", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, pp. 261-268

[Khos89b] Pradeep K. Khosla, "Real-Time Implementation and Evaluation of Computed-Torque Scheme", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 2, pp. 245-253

[Koiv91] A-J Koivo and M-A. Unseren, "Reduced Order Model and Decoupled Control Architecture for Two Manipulators Holdinga Rigid Object.", 1991, Transactions-of-the-ASME-Journal-of-Dynamic-Systems-Measurement-and- Control (USA), Vol. 113, No. 4, pp. 646-54

[Kokk92] Théodore Kokkinis and Pierre Millies, "Kinetostatic Performance of a Dynamically Redundant Parallel Robot", 1992, International-Journal-of-Robotics-Automation (USA), Vol. 7, No. 1, pp. 30-37

[Kokk91a] Théodore Kokkinis, Yoshihiko Nakamura and Darrin Uecker, "Dynamic Considerations and Motion Control Experiments for a Parallel Direct-Drive Robot Arm", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1118-1124

[Kokk91b] Théodore Kokkinis and R. Stoughton, "Dynamics and Control of Closed-Chain Robot Arms with Application to a New Direct-Drive Robot Arm.", 1991, International-Journal-of-Robotics-Automation (USA), Vol. 6, No. 1, pp. 25-34

[Kreu88] K Kreutz and A. Lokshin, "Load Balancing and Closed-Chain Multiple Arm Control.", 1988, Proceedings of Proceedings of the 1988 American Control Conference, Atlanta, GA, USA, Vol. 3, No. 3, pp. 2148-55

[Kreu89] K. Kreutz, "On Manipulator Control by Exact Linearization", 1989, IEEE Trans. Automatic Control, Vol. 34, No. 7, pp. 763-767

[Leah89] M.B. Leahy, "*Experimental Analysis of Model-Based PUMA Robot Control*", 1989, Air Force Institute of Technology, Robotic Systems Laboratory, Ohio Technical Report ARSL-89-3

[LeeG86] C.S. George Lee and P.R. Chang, "Efficient Parallel Algorithm for Robot Inverse Dynamics *Computation*", *1986, IEEE Trans. on Systems, Man and Cybernetics*, Vol. 16, pp. 532-542

[LeeG87] C.S. George Lee and P.R. Chang, "A Maximum Pipelined CORDIC Architecture For Inverse Kinematic Position Computation", 1987, RA, Vol. 3, No. 5

[LeeK88a] Kik-Meng Lee and Dharman K. Shah, "Kinematic Analysis of a Three-Degrees-of-Freedom In-Parallel Actuated Manipulator", 1988, IEEE Journal of Robotics and Automation, Vol. 4, No. 3, pp. 354-360

[LeeK88b] Kik-Meng Lee and Dharman K. Shah, "Dynamic Analysis of a Three-Degrees-of-Freedom In-Parallel Actuated Manipulator", 1988, IEEE Journal of Robotics and Automation, Vol. 4, No. 3, pp. 361-367

[Li88] Chang-Jin Li, "A Fast Computational Method of Lagrangian Dynamics for Robot Manipulators", 1988, International Journal of Robotics & Automation, Vol. 3, No. 1, pp. 14-20

[LinS90] S-K. Lin, "Dynamics of the Manipulator with Closed Chains.", 1990, IEEE-Transactions-on-Robotics-and-Automation (USA), Vol. 6, No. 4, pp. 496-501

[Liu91] K. Liu et al., "Dynamic and Kinematic Study of a Stewart Platform using Newton-Euler Techniques", 1991, Transactions of the ASME Symposium on Control of Systems with Inexact Dynamic Models, Atlanta, USA, pp. 83-89

[Liu93] K. Liu, F. Lewis, G. Lebret and D. Talor, "The Singularities and Dynamics of a Stewart Platform Manipulator", 1993, J. of Intelligent Robotic Systems, No. 8, pp. 287-308

[Luh80a] J.Y.S. Luh, M.W. Walker and R.C.P. Paul, "Resolved-Acceleration Control of Mechanical Manipulators", 1980, IEEE Transactions on Robotics and Automation, Vol. 25, No. 3

[Luh80b] J.Y.S. Luh, M.W. Walker and R.C.P. Paul, "On-line Computational Scheme for Mechanical Manipulators", 1980, ASME Trans. Journal on Systems, Meaurements and Control, Vol. 102, p. 468-474

[Luh85] J.Y.S. Luh and Yuan F. Zheng, "Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms", 1985, IEEE Journal of Robotics and Automation, Vol. 1, No. 2, pp. 95-103

[MaO91] O. Ma and J. Angeles, "Optimum Architecture Design of Platform Manipulators", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1130-1135

[MaXF88] Xiang-Feng Ma and Xiang-Rong. Xu, "A Further Study on Kane's Equations Approach of Robots Dynamics.", 1988, Proceedings of Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, Beijing and Shenyang, China, Vol. 1, No. 2, pp. 107-12

[MaXF90] Xiang-Feng Ma, Xiang-Rong Xu, B Ed. by Svrcek and J. McRae, "A New Efficient Algorithm for Robots Dynamics.", 1990, Proceedings of Proceedings of the 1990 Summer Computer Simulation Conference, Calgary, Alta., Canadapp.300-4

[Mark74] B.R. Markiewicz, "Analysis of the Computed-torque Drive Method and Comparison with Conventional Position Servo for a Computer Controlled Manipulator", February 1974, Jet Propulsion Labreport JPL TM 33-669

[Merl88] J-P. Merlet, "Force-Feedback Control of Parallel Manipulators.", 1988, Proceedings of Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, Vol. 3, No. 3, pp. 1484-9

[Merl89] J-P. Merlet, "Singular Configurations of Parallel Manipulators and Grassmann Geometry.", 1989, International-Journal-of-Robotics-Research (USA), Vol. 8, No. 5, pp. 45-56

[Merl90] *Jean-Pierre Merlet, "Les robots parallèles", Hermès, Paris, 1990*

[Merl91] Jean-Pierre Merlet, "An Algorithm for the Forward Kinematics of General Parallel Manipulators", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1136-1140

[Midd88] R.H. Middleton and G.C. Goodwin, "Adaptive Computed Torque Control for Rigid Link Manipulations", 1988, Systems and Control Letter, Vol. 10, pp. 9-16

[Mill92] K. Miller and R. Clavel, "The Lagrange-Based Model of Delta-4 Robot Dynamics", 1992, Robotersysteme, Vol. 8, pp. 49-54

[Murr87] J-J Murray and G-H. Lovell, "Closed Chain Manipulator Dynamics and Control.", 1987, Proceedings of Proceedings of the Twenty-Fifth Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, Vol. 2, pp. 829-38

[Naka89] Y. Nakamura and M. Ghodoussi, "Dynamics Computation of Closed Link Robot Mechanisms with Nonreduandant and Reduandant Actuators.", 1989, IEEE Journal on Robotics and Automation, Vol. 5, No. 3, p. 294-302

[Neum85] Charles P. Neuman *and Vassilios D. Tourassis*, "Discrete Dynamic Robot Models", 1985, SMC, Vol. SMC-15, No. 2, pp. 193-204

[Niem88] G. Niemeyer and J.-J. E. Slotine, "Performance in Adaptive Manipulator Control", 1988, Proceedings of Int. Conf. on Decision and Control, Austin, USA

[Orte89] Romeo Ortega and Mark W. Spong, "Adaptive Motion Control of Rigid Robots: a Tutorial", 1989, Automatica, Vol. 25, No. 6, pp. 877-888

[Pade88] Brad Paden and Ravi Panja, "Globally Assymptotically stable 'PD+' Controller for Robot Manipulators", 1988, Int. Journal of Control, Vol. 47, No. 6, pp. 1697-1712

[Paul81] Richard P. Paul, "Robot manipulators: Mathematics, Programming, and Control", MIT, 1981

[Penn92] *G.R. Pennock and B.A. Oncu, "Application of Screw Theory to Rigid Body Dynamics"*, 1992, Transactions of the ASME

[Pete92] Steven Peterson, "The Geometry of Virtual Work Dynamics in Screw Space", 1992, *Transactions of the ASME*

[Pier91] F. Pierrot, P. Dauchez and A. Fournier, "HEXA: a Fast Six-DOFs Fully-Parallel Robot", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1158-1163

[Pier92] F. Pierrot, P. Dauchez and A. Fournier, "Towards a Fully-Parallel 6 DOF Robot for High-Speed Applications", 1992, Int. Journal on Robotics and Automation, Vol. 7, No. 1, pp. 15-22

[Pier92] F. Pierrot, P. Fraisse, X. Delebarre and P. Dauchez, "Manipulations robotiques à haute vitesse : une solution pleinement parallèle", 1992, RAIRO APII Robotique Industrielle, Vol. 26, No. 1, pp. 1158-1163

[Poll38] *W.L.V. Pollard, "Position Controlling Apparatus", US patent, April 1938*

[Rebo90] Claude Reboulet, "Nouvelles architectures : les robots parallèles", 1990, Assemblages, Vol. 120, pp. 27-30

[Rebo91] Claude Reboulet and Thierry Berthomieu, "Dynamic Models of a Six Degrees-of-Freedom Parallel Manipulator", 1991, Proceedings of ICAR, Pisa, Vol. 2, No. 2, pp. 1153-1157

[Reed89] John S. Reed and Petros A. Ioannou, "Instability Analysis and Robust Adaptive Control of Robot Manipulators", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, pp. 381-386

[Sada89] Ponnuswamy Sadayappan, Yong-Long Calvin Ling and Karl W. Olson, "A Restructurable VLSI Robotics Vector Processor Architecture for Real-Time Control", 1989, IEEE Transactions on Robotics and Automation, Vol. 5, No. 5, Special Issue on Algorithms and Architecture, pp. 583-599

[Sark89] Vivek Sarkar, "Partitioning and Scheduling Parallel Programs for Multiprocessors", Pitman, London, 1989

[Schw90] H.M. Schwartz, G. Warshaw and T. janabi, "Issues in Robot Adaptive Control", 1990, Proceedings of the Automatic Control Conference, pp. 2797-2805

[Silv82] W.M. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", 1982, Int. J. of Robotics Research, Vol. 1, No. 2, pp. 60-70

[Slot83] J.-J. E. Slotine and S. S. Sastry, "Tracking Control of Nonlinear Systems using Sliding Surfaces with Application to Robot Manipulators", 1983, Int. Journal of Control, Vol. 38, No. 2, pp. 485-492

[Slot85] J.-J. E. Slotine, "The Robust Control of Robot Manipulators", 1985, Int. Journal of Robotics research, Vol. 4, No. 2

[Slot87a] J.-J. E. Slotine and W. Li, "On the Adaptive Control of Robot Manipulators", 1987, *Int. Journal of Robotics Research, Vol. 6, No. 3*

[Slot87b] J.-J. E. Slotine and W. Li, "Adaptive Manipulator Control : a Case Study", 1987, Proceedings of Int. Conf. on Robotics and Automation, Raleigh, USA

[Slot91] J.-J. E. Slotine and W. Li, "Applied Nonlinear Control", 1991, Prenctice-Hall.

[Spon89] *Mark W. Spong and M.Vidyasagar, "Robot Dynamics and Control", John Wiley & Sons, New* York, 1989

[Spon90] *Mark W. Spong and Romeo Ortega, "On Adaptive Inverse Dynamics Control of Rigid* Robots", 1990, IEEE Transactions on Automatic Control, Vol. 35, No. 1, pp. 92-95

[Ster87] F. Sternheim, "Computation of the Direct and Inverse Geometric Models of the Delta4 Parallel Robot", 1987, RoboterSysteme, Vol. 3, pp. 199-203

[Stew65] D. Stewart, "A platform with 6 Degrees-of-Freedom", 1965, Proc. of the Institute of mechanical engineers, Vol. 180, Part 1, No. 15, pp. 371-386

[Sugi87] K. Sugimoto, "Kinematic and Dynamic Analysis of Parallel Manipulators by Means of Motor Algebra", 1987, Transactions of the ASME, Journal of Mechanisms, Transmissions, and *Automation in Design, Vol. 109, No. 1, pp. 3-7*

[Sugi89] K. Sugimoto, "Computational Scheme for Dynamic Analysis of Parallel Manipulators.", 1989, Transactions-of-the-ASME-Journal-of-Mechanisms-Transmissions-and-Automation-in-Design (USA), Vol. 111, No. 1, pp. 29-33

[Tarn91a] T-J Tarn, A-K Bejczy, X. Yun and Z-F. Li, "Effect of Motor Dynamics on Nonlinear Feedback Robot Arm Control", 1991, IEEE Transactions on Robotics and Automation, Vol. 7, No. 7, pp. 114-122

[Tarn91b] T.J. Tarn, G.T. Marth, A.K. Ramadorai and A.K. Bejczy, "Six Degree-of-Freedom Task *Space Control For the PUMA 560 Manipulator : An Experimental Study", 1991, Proceedings* of SYROCO, Vienna, Austria, Vol. 1, No. 2, pp. 177-182

[Tarn93] T.J. Tarn, A.K. Bejczy, G.T. Marth and A.K. Ramadorai, "Performance Comparison of Four Manipulator Servo Schemes", 1993, IEEE Control Systems, Vol. February 1993, pp. 22-29

[Tour85] Vassilios D. Tourassis and Charles P. Neuman, "Inverse Dynamics Applications of Discrete Dynamic Robot Models", 1985, SMC, Vol. SMC-15, No. 6, pp. 798-803

[Tour89] Vassilios D. Tourassis and Marcello H. Ang, "A Modular Architecture for Inverse Robot Kinematics", 1989, IEEE Trans. on Robotics and Automation, Special Issue on Algorithms and Architecture, Vol. 5, No. 5, pp. 555-568

[Tso 93] S.K. Tso and L.Y.X. Ma, "Discrete learning control for robots: strategy, convergence and robustness", 1993, Int. Journal of Control, Vol. 57, No. 2, pp. 273-291

[Unda87] J. Unda, J. Garcia de Jalon, F. Losantos and R. Enparantza, "A Comparative Study on Some Different Formulations of the Dynamic Equations of Constrained Mechanical Systems", 1987, Transactions of the ASME

[Vill84] G. Villalonga, J. Unda and J. Garcia de Jalon, "Numerical Kinematic Analysis of Three-Dimensional Mechanisms Using a 'Natural' System of Lagrangian Coordinates", 1984, ASME Paper, Vol. 84-DET-199

[Walk82] M.W. Walker and D.E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", 1982, Trans. ASME Jnl. Dynamic Systems Measurement and Control, Vol. 104, pp. 205-211

[Walk90] Michael W. Walker, "Adaptive Control of Manipulators Containing Closed Kinematic Loops", 1990, IEEE Transactions on Robotics and Automation, Vol. 6, No. 1, pp. 10-19

[Wand87] John. P. Wander and Delbert Tesar, "Pipelined Computation of Manipulator Modeling Matrices", 1987, IEEE Transactions on Robotics and Automation, Vol. RA-3, No. 6, pp. 556-566

[Wen 88] John T. Wen and David S. Bayard, "New Class of Control Laws for Robotic Manipulators. Part 1 : Non-adaptive case.", 1988, Int. Journal of Control, Vol. 47, No. 5, pp. 1361-1385

[Youn78] K.K.D. Young, "Controller Design for a Manipulator using the Theory of Variable Structure Systems", 1978, IEEE trans. Systems, Man and Cybernetics, Vol. SMC-8, pp. 210-218

[Zai92] L.C. Zai, L.F. Durfee, D.G. Manzer, J.P. Karidis, M.P. Mastro and L.W. Landerman, "Control of a Hummingbird Micropositionner with a Multi-transputer MARC Controller", 1992, Proceedings of Int. Conf. on Robotics and Automation, Nice, France, pp.534-541

[Zhan91] H Zhang and R-P. Paul, "A Parallel Inverse Kinematics Solution for Robot Manipulators Based on Multiprocessing and Linear Extrapolation.", 1991, IEEE-Trans-Robotics-Automation, Vol. 7, No. 5, pp. 660-669

[Zoma90] A.Y. Zomaya and A.S. Morris, "Modeling and Simulation of Robot Dynamics using Transputer-Based Architectures", 1990, Simulation, Vol. may 1990, pp. 269-278

[Zoma92] A.Y. Zomaya and A.S. Morris, "Parallel Computation of Robot Inverse Dynamics for High Speed Motions", 1992, IEE-Proceedings-D, Vol. 139, No. 2, pp. 226-236

Curriculum Vitae.

Nom:	GUGLIELMETTI
Prénom:	Philippe
Né le:	25 décembre 1963
Origine:	Morbio Superiore, Tessin

Formation.

- Maturité type C, Collège de St-Maurice, Valais 1979-1983.
- 1ère année en Section de Microtechnique à l'Ecole Polytechnique Fédérale de Lausanne (EPFL)
  3 années suivantes en Section Informatique, orientation matériel.
- Diplôme d'ingénieur informaticien EPFL en 1988.

Activités professionnelles.

- Assistant à l'Institut d'Automatique de l'EPFL de 1988 à 1994.
- Projet industriel pour SOCAPEL SA, Penthaz en 1988-1989. Réalisation d'un logiciel d'identification automatique des paramètres de l'entraînement électrique et de synthèse du régulateur RST par placement des pôles en boucle fermée.
- Développement d'un noyau temps réel pour la commande multiprocesseurs (68000) d'un robot hydraulique expérimental (1989-1990).
- Travail de recherche sur l'utilisation des réseaux de neurones pour la commande dynamique des robots. Présentation des résultats à l'IFAC Workshop on Intelligent Motion Control, Istanbul, 1990.
- Mandat de CGS Concept SA, Lausanne. Développement du logiciel pour le wagon de mesure des chemins de fer autrichiens (OEBB) suite au décès du responsable du projet (1992).
- Etude de la dynamique des robots parallèles rapides et réglage avancé, avec application au robot DELTA. Ce travail a été financé par le Fonds National Suisse de la Recherche Scientifique et a débouché sur une thèse. Résultats présentés à l'IFAC Workshop on Motion Control for Intelligent Automation, Perugia, 1992.