AN ANALYTICAL CONSIDERATION OF MULTI BODY DYNAMICS AS APPLIED TO ROBOTIC STRUCTURES

C. VAN DER MERWE

An Analytical Consideration of Multi Body Dynamics as applied to Robotic Structures

by

Carel van der Merwe

A dissertation submitted towards the fulfillment
of the requirements for the degree

Magister Ingeneriae

in

Mechanical Engineering

in the

Faculty of Engineering

of the

Rand Afrikaans University

Supervisor: Prof. André L. Nel

November 1995

Abstract

This dissertation presents and explains methods for the dynamic modeling of robot like mechanisms. These mechanisms can have multiple degree of freedom joints and contain closed-chains.

A new kinematic notation is proposed. The algorithms, including those used for the inverse and direct dynamics, are all based on spatial notation and a general joint model, providing a quite general, integrated and complete method for solving the dynamics of simple closed-chain mechanisms.

Opsomming

Hierdie verhandeling toon en verduidelik metodes om die dinamika van robot tipe meganismes te modelleer. Hierdie meganismes kan skarniere met veelvoudige grade van vryheid bevat, asook geslote kettings.

'n Nuwe kinematiese notasie word voorgestel. Die algoritmes, insluitend die vir die inverse en direkte dinamika, maak almal gebruik van ruimtelike notasie en 'n algemene skarniermodel, wat 'n redelik algemene, geïntegreerde en volledige metode daarstel vir die oplos van die dinamika van eenvoudige geslote-ketting meganismes.

Contents

List of Figures

Chapter 1

Introduction

Therefore, O students, study mathematics, and do not build without foundations.

Leonardo da Vinci
Quaderni d'Anatomia I $13v.1$

No matter how much we abstract our world or how deep we delve into the depths of Cyberspace, virtual reality, finance and economy, the human race can never escape the fact that its existence and survival is irrevocably dependent on its interaction with the physical world we live in (in this world, at least). It is a world where objects have mass, energy and chemical and nuclear properties. By interacting with this world, we must find food, shelter, and companions.

There will always be the need to manipulate physical objects, which in the most basic form would be to move objects in space. This is exactly what we do when we sow seeds in order to raise crops, put one brick on top of another to build a house, assemble parts to manufacture a car, or walk/sail/drive/fly to the vicinity of another human being in order to interact with it.

As we abstract our world, human physical labour becomes less acceptable – socially and economically. Even now, it is frequently better to let machines do the physical work. It might not only be cheaper, more accurate and reliable, but it frees workers from boredom in repetitive work. Often it is morally unacceptable to let humans perform certain tasks due to the danger involved in it. In such instances one has no option but to use a mechanical manipulator.

Examples where such manipulators, or robots, are used, include the following:

- In industry:

  - Spray painting
  - Welding
  - Materials handling and warehousing
  - Machine loading
  - Assembly
  - Measurement
  - Handling of toxic waste
  - Cleaning the pipes of nuclear power plants

- In science:
  - Manipulating equipment in space
  - Performing hazardous experiments
  - Assembly in clean rooms
  - Multi-legged vehicles

- In medicine:
  - Precision surgery
  - Actuated prostheses

The list of current and future applications involving robotic systems might sound like science fiction, but the reality remains that we are dependent on systems that can manipulate objects, and we will become even more dependent on such systems in the future.

Such systems usually consist of jointed structures – to allow the motion of one part with respect to another – and actuators to exert a force, in a certain direction, on the parts joined at the joint. Some control system is also required to control the forces and ultimately the positions of the parts of the structure. Not all systems of this type are associated with such glamorous applications as cited above, but common cranes on construction sites and on quays, and machines in a production process which fills a container with some substance or which sticks a label on an assembly, also forms part of these systems.

Such systems must be designed – both mechanically and electronically, and their control systems must be actualized – then built, tested and operated. At the conceptual design stage, the designers must know what the system must do, in this case, how it must move and how it must manipulate the environment. They must have a good idea of what forces the structure will experience in operation. These forces can be measured after the system has been built, but it is advisable and more economic to design the system before building it. In order to determine the behaviour of the system before it is built, one needs to construct a mathematical model of the system. Both the mechanical behaviour of the structure and the working of the control system can be modeled mathematically. The model for the mechanical behaviour is called the dynamic model[1] of the structure, and can be used to determine what forces the structure will experience if moving in a certain way, and how it will move if certain forces are applied to it. The designers can then optimize these models and iteratively reach an acceptable concept.

When the mechanical design has matured and a concept control system has been designed, the control system can be tested by using its control outputs as inputs to the dynamic model, and the states of the dynamic model as inputs to the control system. Immediately a distinct advantage of a dynamic model should be clear: all the state variables are at hand. In a physical system it might be very difficult and impractical to measure all the state variables. This should aid control designers considerably in optimizing the performance of the system.

If the control system designer happened to choose some form of adaptive control, this control system can be trained on the dynamic model of the system, before installing it in the physical system. In this way they can avoid possible damage to the mechanical structure of the system, caused by a control system that cannot yet adequately control the physical system. For very difficult control tasks, training the controller on the mathematical dynamic model may be the only way of doing it because of the very large number of training iterations needed.

In tele-operated systems[2] – where the controller or operator is so far from the controlled system that there is an appreciable time delay for the closing of the feedback loop, it might not be possible to control the system based on its state a few seconds ago. In this case one would have to simulate how the system is expected to react, and at any time base the present control actions on the estimated state of the system at the future point – as given by the simulation. The simulation can then itself be fine tuned when the state feedback arrives, to make it more accurate.

From the discussion it can be seen that a mathematical model for the dynamic

[1] A model that relates the general forces on a manipulator to the first and second time derivative of the general position of the structure. The general position can either be the position of the end effector, or the joint positions of the joints in the manipulator.

[2] A manipulator on a spacecraft that is being controlled by an operator on earth is an example of a tele-operated system.

(mechanical) behaviour of a manipulation system is helpful or even indispensable in:

1. the design of the mechanical structure of manipulators and similar equipment;
2. the design of control systems for such equipment;
3. the testing of such conceptual systems complete with a controller;
4. the control of tele-operated systems;

and it has been shown how such a model fit into the greater scheme of things in the struggle for survival and improvement of quality of live, of the human race.

1.1 Dynamics of robotic structures and mechanics in general

The investigation of dynamic behaviour of robot manipulators forms part of the general field of *Mechanics* (see Figure 1.1). As described by Truesdell [1], (Rational) Mechanics is the part of mathematics that provides and develops logical models for the enforced changes of position and shape which we see everyday things suffer.

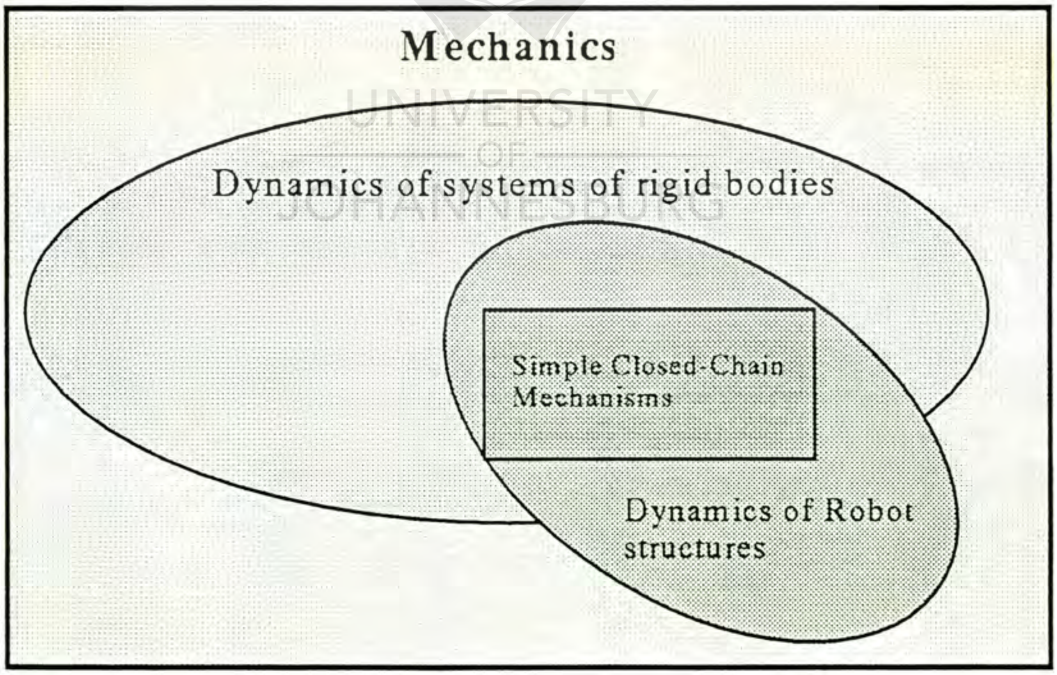

Mechanics
Dynamics of systems of rigid bodies
Simple Closed-Chain Mechanisms
Dynamics of Robot structures

Figure 1.1: The field of robot dynamics in mechanics.

Mechanics does not study natural and man made things directly, but instead considers *bodies.* Bodies are mathematical concepts that are designed to abstract common features of natural things. The features that are abstracted can include the geometry, motion (kinematics), inertial properties, forces, elastic properties, and energy of bodies. Mechanics is thus a finite class of mathematical models for certain aspects of nature. Kinematics forms an important part of the foundation of mechanics.

For a clarification of the concepts of bodies, the event world, frames of reference, motions, forces and energies, the reader is referred to the first chapter of [1].

Within the field of mechanics, one finds the branch concerned with the study of rigid body systems. The system consists of a set of bodies, as defined by the analyst, whose behaviour is investigated. Everything external to this set is the environment of the system. The bodies, which are the objects of this set, are objects whose properties have been abstracted to spatial geometry, inertial properties, and forces acting on them. The deformation of the bodies are neglected. This is a small branch of mechanics in general. Because of the exclusion of deformation and energy (and thus temperature) from the properties of bodies, the substantial branches of mechanics concerned with the study of thermodynamics and fluid dynamics is excluded from this branch.

In robotic structures (see Figure 1.2), the deformation of the link can often be neglected, depending on the structure itself and the object of the modeling of this structure. If this can be done, the manipulator can be viewed as a special case of a system of rigid bodies. This dissertation is exclusively concerned with the mathematical modeling of such systems with the object of making predictions on their behaviour.

If the structure of a manipulator is such that it is relatively flexible and that the deflection of its links under working conditions can have a significant effect on the position of the end effector, this manipulator can not be treated as a system of rigid bodies. In such cases the analyst would often be concerned with the vibrations of the structure. Such problems are still part of mechanics, but not of the specialized field that this dissertation is concerned with.

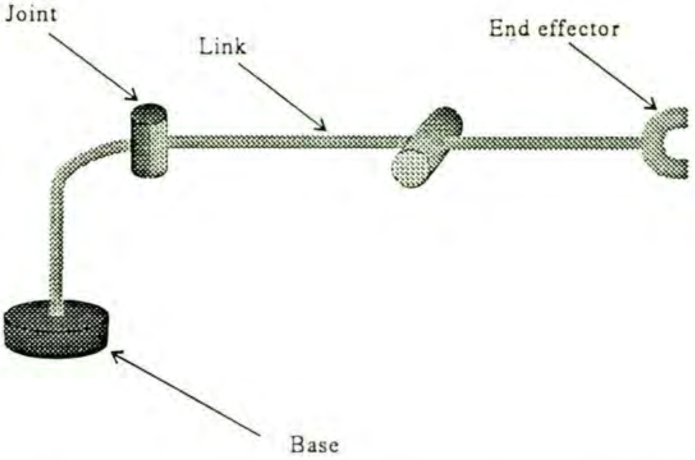


Figure 1.2: The main components of a robot manipulator.

1.2 The dynamics of robotic manipulators in particular

In the dynamics of systems of rigid bodies in general, the relationship between generalized positions, forces and torques, and time is usually sought. This usually involves describing the relations between forces and torques and the second time derivative of the generalized positions. Time is thus treated as the independent variable[3].

Even at this stage, it can be seen that the problem can be viewed from two perspectives: the forces and torques are known and the accelerations are sought, or the required accelerations are known and the torques and forces needed to achieve this motion is sought. These two problems are termed the *direct dynamics* and the *inverse dynamics* problem. The direct dynamics problem is usually encountered when simulations are done, while the solution to the problem is often used in the development of control systems. It will later be shown that elements of both solutions can be used to advantage in either of the problems [2].

In this dissertation, a robotic manipulator (see Figure 1.2) will be defined as a system of interconnected rigid bodies. The bodies are connected by *hinges* which can transmit a combination of forces and torques between the bodies they connect. These hinges[4] can also constrain the relative motion of these bodies with respect to

[3] The validity of this philosophy will not be discussed in any depth.

[4] The hinges need not be "physical" hinges, bodies can interact with each other through gravitational, electrostatic, electro-magnetic and other forces and constraints. The effect of all these interactions will be represented by hinges between bodies.

each other.

The interactions between any pair of bodies will always be represented by a single hinge or joint. Any body can however have more than one hinge, that is, it can interact directly with more than one other body. In many mechanical systems one encounters a body whose motion is either prescribed or not influenced seriously by the other bodies, although it influences them (eg. a very massive body like earth when the movement of small bodies on its surface is considered). This body will be termed the *base*.

How the interconnection of bodies is described, is determined by the structure of the system. If all of the bodies in the system has at most two hinges, the system is described as a *serial chain* of bodies. Most robotic manipulators fall into this class. If only one body in this serial chain interacts with the base, this class is called open chain mechanisms. The manipulator illustrated in Figure 1.2 is an example of a open serial chain. If the two bodies on the opposite ends of a serial chain is connected to a base (not necessarily the the same base), this is called a closed chain mechanism. This is often encountered when the end-effector of a manipulator makes contact with its environment, as illustrated in Figure 1.3. The interconnection of bodies in a serial chain can simply be described by numbering bodies consecutively, starting at a base. Some confusion can arise as to the effect of gravity. The earth can be seen as a body that interacts with all the bodies in the system, by exerting a force on them all. This approach unnecessarily complicates analyses, as no serial chain with more than one body in a gravitational field would exist. The effect of gravity can rather be seen as giving the reference frames fixed relative to earth an upward acceleration, which of course makes them *non-inertial frames*[5].

When the system has at least one body with more than two hinges, the mechanism is *branched*. Such a system is also called a multiple chain mechanism, because the branches can each be seen as a serial chain. See Figure 1.4. If the structure is such that one can move from one body to the other across hinges in one direction and visit one or more bodies more than once, there are *loops* in the system. This is illustrated in Figure 1.5. For such systems as the above, also called systems with *tree structure*, the interconnection structure can be described by interconnection matrices, as introduced by Wittenburg [3].

A multiple closed-chain mechanism is a branched mechanism with more than two

[5] A frame of reference in which the Newton-Euler laws of motion appear not to be valid if the variables are measured in this frame. A frame that has an acceleration or angular velocity relative to an inertial frame, will be a non-inertial frame.

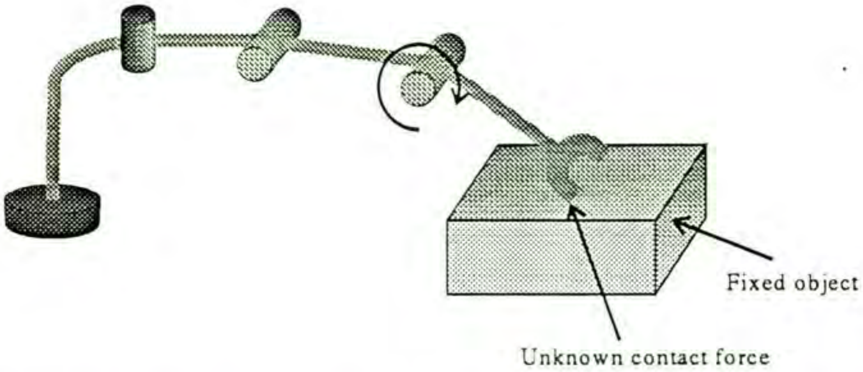


Figure 1.3: A robot manipulator that is in a serial closed chain configuration.

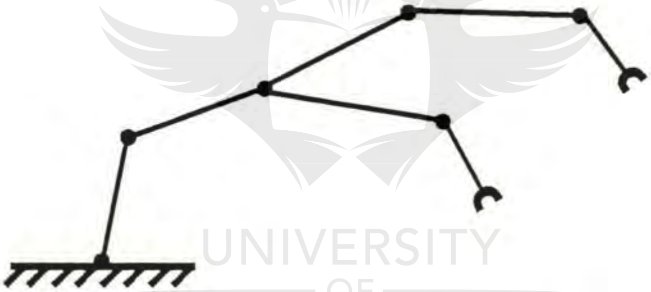


Figure 1.4: A multiple chain mechanism.

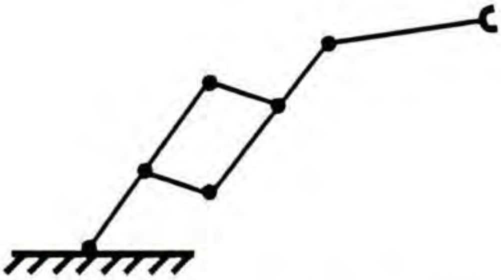


Figure 1.5: A mechanism with one internal loop.

of the bodies in contact with a base. If all the closed chains in the system can be broken by the removal of a single body and the chains that remain are all serial, the original mechanism is called a simple closed chain mechanism.

1.3 Dynamic formalisms

A dynamic formalism is a set of equations that describe the laws of nature that relate to the dynamics of a single body. Historically several formalisms have been developed. They include:

- Vector formulation of Newton's second law together with Euler's equation for rotational motion [3].
- The Lagrange-Euler formulation [4].
- The Recursive Lagrangian formulation [5].
- The Generalized D'Alembert formulation.
- Hamilton equations.
- Kane's equations [6].
- Appel's equations [7].

All of these formalisms have been applied successfully to multi-body systems. The Newton-Euler equations are perhaps the most widely used and easiest to give a physical interpretation to the results of the analyses.

1.3.1 Choice of formalism

The Lagrange-Euler and Newton-Euler equations have been used most extensively in the robotics field. For years there has been a debate as to which formalism leads to the most efficient algorithms. Intuitively, one would expect all the formalisms to lead to the same or equivalent algorithms and solutions.

The debate was finally settled by Silver [8], who showed that all the formalisms can lead to the same algorithms. The difference between the efficiencies are largely due to the different representations of rotational motion. Balafoutis and Patel [9] also shows how the same algorithm for the inverse dynamics can be derived by using either the Newton-Euler, the Euler-Lagrange or Kane's formalisms.

The choice of which formalism to use is therefore largely a matter of personal taste, although many people prefer the Newton-Euler formalism for its simple equations and direct physical interpretation.

1.4 Efficiency of simulation algorithms

Because solving for the dynamic behaviour of a multi-body system involves numerous calculations, the efficiency of the proposed algorithm is very important when *real-time* simulation is sought. It is generally difficult to do real-time simulation of even simple systems without resorting to prohibitively expensive computers[6]. Because of this there is an ongoing quest to the improvement of the computational efficiency of algorithms.

It should be realized that there may be other considerations that outweigh the need for computational efficiency, for example ease of application, simplicity and generality, and which may be more important today because of the increased performance of even low cost computers.

The computational efficiency of an algorithm is greatly influenced by the representation of the physical and kinematic quantities appearing in the equations of motion (for example inertia, rotational motion etc.). In the Euler-Lagrange formalism these quantities are frequently expressed as sets of scalars, while the representation of kinematic and dynamic quantities as vectors is more natural in the Newton-Euler formalism. The use of angular velocity in particular contributes greatly to the efficiency of the Newton-Euler derived algorithms. Balafoutis and Patel [9] has shown that the representation of many of these quantities as Cartesian Tensors is even more efficient. Further, the computational efficiency is also influenced by the modeling scheme used. The modeling scheme used by most algorithms considers each body as a separate rigid body and the chain as an ideally connected chain of rigid bodies. Another modeling scheme is to use the concept of generalized and augmented bodies [2, 3, 9].

Whether the algorithms are formulated using explicit or recursive equations perhaps has the greatest influence on computational efficiency. Often recursive formulations are more efficient. When *recursive* equations are used, there are intermediate variables involved. These intermediate variables can be chosen in such a way so as to minimize recalculations and the number of calculations involving intermediate

[6] With the current rate of change in electronic technology, affordable computers with the necessary speed can soon be a reality. In that case one can argue that computationally efficient algorithms should still be sought so as to avoid the unnecessary increase of entropy in the universe

variables. This ties in closely with the concept of organizing the structure of computations in the most efficient way.

When the Newton-Euler formalism is used, the choice of in which frame of reference to express the dynamic equations can also influence the efficiency of the algorithm. Expressing the equations for the rigid bodies in the chain in their respective body fixed frames, greatly improves the efficiency of the resulting algorithm. This step has been taken by Luh, Walker and Paul [10].

Customization involves the simplifying of algorithms for special cases such as manipulators with only a single degree of freedom *revolute joint* or 90^o angle of twist. This results in considerable computational savings, but at the cost of a loss in generality. The computational time can also be reduced by implementing parallel processors and optimizing the algorithms for such implementations [2, 11, 12, 13, 14]. Frequently different formulations of the equations of motion and other formalisms are used. Formulations that are efficient for serial computation are not very useful for parallel computation, but can be reformulated to improve efficiency, although formulations developed for parallel implementation from the onset can be much more efficient [14].

1.5 Reference frames

For numeric simulation, kinematic quantities must be measured with respect to some reference frame.[7] Such a quantity is then represented by the components of a coordinate matrix.

Many different frames can be used, such as rectangular Cartesian, cylindrical, spherical and curvilinear systems. If Newton's second law is obeyed in a frame, such a frame is called an inertial reference frame. If the frame has an acceleration or angular velocity relative to an inertial frame, it is a non-inertial frame. Early in the development of the dynamics of multi-body systems, the equations for each body in the system were expressed in inertial reference frames. The more efficient modern approach is to fix coordinate frames to each body in the system and to express the dynamic equations in these non-inertial frames [10]. Expressing the equations in these body-fixed frames is computationally more efficient, but virtual forces (Centripetal and Coriolis) appear in the related dynamic equations. There are no such forces in the equations if the equations are expressed in an inertial frame.

[7] Keep in mind that all the kinematic and dynamic quantities and variables exist even when no reference frame is defined.

The three dimensional space with the definition of the *dot product* as *inner product*, is a nonzero inner product space. It can be proven that every nonzero finite-dimensional inner product space has an orthonormal basis [15]. To simplify descriptions, the coordinate frames used will be orthonormal Cartesian systems. The alternative curvilinear coordinate type systems lead to complex functions that relate kinematic quantities to the generalized coordinates.

1.6 Representation of positions, velocities and transformations between coordinate frames

1.6.1 The position of a point in space

The position of a point in space relative to a reference frame is always expressed as a function of some generalized coordinates. The complexity of this function depends on the kind of reference frame used. The functions in curvilinear frames are more complicated than those in linear frames. In the robotics field the reference frames used are usually 3D Euclidean space frames (rectangular, cylindrical or spherical Cartesian). The position of a point is thus completely specified by point coordinates, which are functions of the generalized coordinates. The transformation of point coordinates between coordinate frames are identical to the transformation of vectors between coordinate frames, therefore vectors can be used to identify points in space.

1.6.2 The configuration of an object in space

With *configuration* is meant the position and orientation of an object. As is well known, an unconnected rigid body has six degrees of freedom. The configuration can thus be described in terms of six generalized coordinates in a reference frame. If this configuration is described by a vector, this vector should have six components and it is not a vector of physical space. The configuration space for a single body is not an Euclidean space and is six dimensional.

By assuming a vector description of this six dimensional configuration quantity, one has to use curvilinear reference frames, which leads to complex functions for relating the generalized coordinates to the configuration.

This can be dealt with in a number of ways. Frequently a vector is used to describe the position of a point on the body and three orientation parameters is used. Commonly used orientation parameters are the Euler angles and the roll, pitch and yaw angles. The three components of the position vector and the three orientation parameters can be seen as the six components of an "orientation vector". This quantity is not a vector as it does not obey the rules of vector transformations.

To implement a system for describing the configuration of a body in space, a coordinate frame is fixed to this rigid body. The position of the body is now specified by the position of the origin of the body-fixed frame relative to a reference frame. The orientation of the body is described by the orientation of the body-fixed frame relative to an intermediate frame that has the same orientation as the reference frame, but with an origin that coincides with the body-fixed frame.

The relative orientation of two Cartesian coordinate frames with a common origin can be described by a linear transformation that maps the base vectors of one base onto the base vectors of the other base [15]. The linear transformation relates the components of a vector in the reference frame to the components of the same vector in the other (body-fixed) frame[8]. If the coordinate systems are orthonormal, the linear transformation will be an orthogonal transformation. Note that orthogonality is defined with respect to a specific definition of the inner product for the vector spaces, in this case the so called dot or scalar product. The linear transformation is proper and represents a rotation. The 3×3 matrix representation of the rotation in a coordinate frame contains six redundant components. Another representation for rotation without any redundancy would be advantageous.

The rotation transformation satisfies the equation $\dot{\mathbf{R}} = \mathbf{\Phi}\mathbf{R}$, where $\mathbf{\Phi}$ is a second order *skew-symmetric* tensor. Orthogonal differentiable tensors also obey this equation. It is not proven here, but the rotation is indeed a orthogonal tensor [9]. The linear transformation that describes relative orientation can be accomplished by taking the dot product with the second order tensor called the *rotation tensor*. From this follows that rotation can be described by the second order rotation tensor. In 3-D Euclidean space, this rotation tensor has a real orthonormal 3×3 matrix representation relative to an orthonormal basis. The entries of this matrix are the direction cosines that relate the axes of the two coordinate systems. A set of nine direction cosines, which satisfy six orthogonality relationships, completely specify the rotation between any two Cartesian systems. The use of the direction cosines in the rotation tensor permits the use of Cartesian coordinate systems in describing the orientation of a rigid body [9].

[8] The transformation can also be seen as relating the components of a vector in some frame to the components of this vector after it has been rotated, but still expressed in the same frame. The rotation is in the opposite direction to the change of orientation from one frame to the transformed frame (if the transformation had been seen as transforming coordinate frames).

The 3×3 matrix representation of the rotation in a coordinate frame contains nine components. If the three-dimensional rotation group can be represented by a set of less than nine parameters, then the linear transformation system is equivalent to a system with less than nine scalar equations, which is a worthwhile simplification. Steulpnagel [16] has shown why it is impossible to have a global 3-dimensional parameterization for the rotation without singular points. In fact, Hopf showed in 1940 that five is the minimum number of parameters needed to represent the rotation in a global one-to-one manner. Steulpnagel points out that the "quaternion method" of parameterizing the rotation group in a one-to-two way, using four parameters is sufficient for practical purposes, and discusses it and some three-dimensional parameterizations as well as Hopf's method.

There exists no one-to-one global representation of the rotation matrix in terms of three generalized coordinates. In practice, a sequence of rotations about axes are used as parameters – which are not generalized coordinates, and these are used as components of an "orientation vector", as mentioned previously. Mathematically, this "orientation vector" is not a valid representation for rotation, since:

1. Rotation is a second order tensor – that is not skew-symmetric – and therefore cannot be represented by a vector.
2. The generalized angle components does not form a vector because the composition of rotations is associative, but not commutative, while the cross-product is not associative and the vector addition is commutative.

The configuration of a rigid body can thus be described by the position vector of its origin and a second order rotation tensor.

1.6.3 The displacement of one body relative to another

In this dissertation, displacement will have the broader meaning of including both the difference between two position vectors for a point on a rigid body in different locations in physical space, and the difference between the two orientations of the same rigid body. These displacements can be described by homogeneous transformations – that may change the orientation and position of a coordinate frame – analyzed in terms of homogeneous coordinates. These are discussed in some detail later. It can also be described by separate linear and rotational displacements, which is more efficient but results in less compact equations.

1.7 Objectives and motivation

The field of robot dynamics is well researched, and solutions to many important problems have been proposed by numerous researchers. However, a practitioner who wants to construct and use a dynamic model of a robotic structure for simulation or control purposes might find it very difficult to do so from the accessible literature. The literature that considers applications is usually of limited generality, while the literature that describes general enough methods lacks some important detail regarding the application of their methods. These details might be obvious to seasoned workers in the field, but to others it can represent great obstacles.

Another problem encountered is that although most problems have been solved, many of them has been solved in isolation, using different kinematic notations, dynamic models and joint models. Integrating the different methods in one complete model is not easy.

The inverse dynamics problem for serial open chains has received a great deal of attention and has progressed to the stage where real-time solutions are possible. Methods based on various dynamic formalisms has been implemented successfully. Recursive methods based on the Newton-Euler formulation are common [10]. The direct dynamics problem for closed-chain serial mechanisms has received less attention, but is also well developed.

The purpose of the author's research and this dissertation is to integrate some of the existing methods and derive new methods where necessary, and to shed some light on the details of applying such methods, in order to propose a scheme for modeling robot dynamics that is reasonably general and easy to apply. Models for both the forward dynamics and inverse dynamics are based on the same notations and modeling scheme. A practitioner should be able to use this scheme to model most robotic mechanisms, including manipulators with multiple degree-of-freedom joints, closed chain mechanisms (as encountered in assembly amongst others), and multiple closed chain mechanisms like walking machines and dextrous hands. The schemes are not customized and takes full account of gravity and other bias forces.

Special attention will be given to developing a kinematic notation that can be used with a joint model [17, 18, 2] that allows up to six degrees-of-freedom joints, and is particularly meaningful from a physical point of view. In many cases, this notation is easier to apply than the Denavit-Hartenberg notation [19].

The numerical implementation of the proposed models will be clarified. The reasons for singularities and the handling of them will be discussed, so as to enable effective simulation of robotic systems.

1.8 Preview

This dissertation presents and discusses methods that can be used to model most robotic mechanisms, specifically with the aim of simulating the behaviour of such mechanisms. The main contents of the remaining chapters are as follows:

Chapter 2 This chapter discusses the forward kinematic modeling of robotic manipulators, and introduces a new kinematic notation that is suitable for use where multi degree of freedom joints are encountered. A suitable joint model is also discussed. Throughout, the spatial notation is used for the configuration analysis.

Chapter 3 This chapter introduces the dynamic aspects of rigid body motion, based on the Newton-Euler formalism. Cartesian tensors are used in this analysis. It will be shown that tensor representations of angular velocity and acceleration can be very advantageous for the Newton-Euler dynamic formulation, and some background on the use of tensors will be included.

Chapter 4 Manipulator inverse dynamics, for manipulators with general joints, is derived in this chapter. It is based on the method of Luh, Walker and Paul [10], but with the general joint model and some Cartesian tensors. An illustrative example of the use of this algorithm is given and discussed.

Chapter 5 A direct dynamics algorithm for single closed chain mechanisms, as used by Lilly [2], is introduced and discussed. A few illustrative examples are given.

Chapter 6 A modeling method for simple closed chain mechanisms is described. The numerical implementation of this method is also discussed in some detail, because singularities are frequently encountered for small systems. An example is also given in the hope of clarifying the use of this method.

Chapter 7 Some aspects of simulating a robotic system are briefly discussed, in particular the interaction of the elements of the simulation and the updating of the configuration of the system.

Chapter 8 The application of the dynamics algorithms to the solution of a rather complex problem is demonstrated and results given.

Chapter 9 Conclusions are presented and some further work that can be done in this field is described.

Chapter 2

Kinematic description of manipulators

Even logic must obey physics.

Dr Spock, USS Enterprise
The Undiscovered Country

2.1 Introduction

To describe the structure of manipulators, we need to describe the relative positions and orientations of the parts of the respective bodies in the manipulator, called *links*. Unlike a general system of rigid bodies, these links cannot move in an arbitrary manner with respect to one another. The links are connected to other links with *joints*, and each of these joints constrain the relative motion of the two links it connects. If we fix a coordinate frame to each link in the manipulator and describe the relative positions and orientations of these frames, we have implicitly also described the structure of the manipulator. Because of the joint constraints, using position vectors and orientation transformations to describe the structure of the manipulator can result in many unnecessary parameters. In this chapter we seek a notation that will conveniently describe the structure of a manipulator. We will consider only serial chains and combinations of serial chains[1] – in practice, these are the most common.

One of the requirements that the structural modeling notation has to fulfill, is that the orientation transformations and position vectors must be obtainable from the parameters in a unique way. However, the inverse is not necessary. It would also be necessary to obtain the relative velocities and accelerations of the links from the

[1] Many complex systems can be divided into serial chains.

time derivatives of these parameters. What modeling notation is deemed as the most suitable will be influenced by how the velocities and accelerations are modeled. In this chapter the representation of such quantities will also be addressed.

2.2 Links, joints and frames

We will use the convention that the body fixed to the *base* is called link 1, and the joint that connects link 1 to the base is called joint 1. The coordinate frame fixed to this body (link) is called frame 1, and is connected to the part of the link closest to joint 1 and thus the base. The other links, joints and frames, are numbered consecutively from link one to the last link in the serial chain. The above can be generalized as: joint n connects link $n-1$ and link n, and coordinate frame n is fixed to the near end of link n. This is illustrated in Figure 2.1.

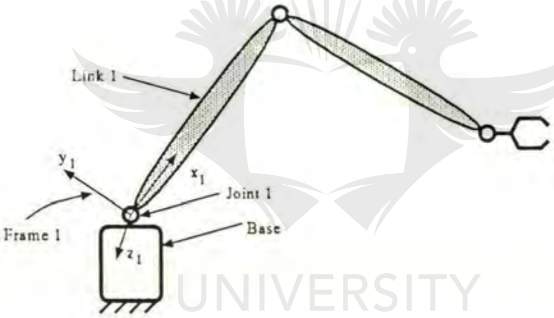


Figure 2.1: Diagram of manipulator to demonstrate the numbering conventions used.

2.3 Representation of kinematic and dynamic quantities

In the rest of this dissertation, the Newton-Euler dynamic formalism will be used. The vector representation of kinematic quantities is well suited for use with this formalism. However, by using the *spatial* representation of these quantities, the equations of motion can be written even more compactly. Apart from this, using this notation has definite advantages in the computation of such quantities as the Jacobian of the manipulator, and makes the use of a simple but quite general joint model possible. Computing the relative velocities of links becomes simple and very compact, as the reader will soon see. The use of this notation is extremely powerful, and the full consequences can only be appreciated after using it for a while.

The theoretical basis for spatial notation is given by Featherstone [20]. Corresponding linear and angular vectors are combined in a single vector. The components of this vector relative to a base is represented by a 6×1 column matrix[2]. Generally the first three rows of a spatial vector are the components of a line vector, and the following three rows are the components of the corresponding free vector. In this way, the spatial velocity of link i can be written as:

$$\mathrm{v_i} = \begin{bmatrix} (\omega_i)_x \\ (\omega_i)_y \\ (\omega_i)_z \\ (v_i)_x \\ (v_i)_y \\ (v_i)_z \end{bmatrix}, \tag{2.1}$$

where $(\omega_i)_x$, $(\omega_i)_y$ and $(\omega_i)_z$ are the components of the angular velocity of link i about $\hat{\mathbf{x}}_\mathrm{i}$, $\hat{\mathbf{y}}_\mathrm{i}$ and $\hat{\mathrm{z}}_\mathrm{i}$, respectively. The other three components represent the linear velocity of the ith coordinate origin, resolved along its own axes. The vector $\mathbf{v}_\mathbf{i}$ is thus expressed in the ith frame. In the same way the spatial acceleration of link i may be expressed as:

$$\mathrm{a_i} = \begin{bmatrix} (\alpha_i)_x \\ (\alpha_i)_y \\ (\alpha_i)_z \\ (a_i)_x \\ (a_i)_y \\ (a_i)_z \end{bmatrix}, \tag{2.2}$$

where the first three rows correspond to the resolved angular acceleration, and the last three rows to the linear acceleration vectors.

The generalized force, $\mathrm{f_i}$, exerted on link i, has the following form:

$$\mathrm{f_i} = \begin{bmatrix} (m_i)_x \\ (m_i)_y \\ (m_i)_z \\ (f_i)_x \\ (f_i)_y \\ (f_i)_z \end{bmatrix}, \tag{2.3}$$

[2] In the following, when referring to a spatial vector, it will also refer to its representation relative to some coordinate frame.

where the first three components represent elements of a three-dimensional moment vector, and the last three are the elements of a three-dimensional force vector. Because of the way in which it is defined, it is not a spatial quantity like velocity and acceleration. Accordingly, it does not satisfy the transformation equation given later (equation 2.5). The definition of a spatial force that does satisfy the general transformation equation, is given by [20] as:

$$\mathbf{f_i} = \begin{bmatrix} (f_i)_x \\ (f_i)_y \\ (f_i)_z \\ (m_i)_x \\ (m_i)_y \\ (m_i)_z \end{bmatrix}. \tag{2.4}$$

Linear force is a line vector and moment is a free vector. In the work that follows, the general force definition (equation 2.3) will be used for uniformity (i.e. the first three components of any 6×1 vector will be associated with rotation).

2.3.1 Transformation of spatial vectors

Similar to ordinary vectors, if the components of a spatial vector relative to some coordinate system is known, its components relative to another coordinate frame can be calculated. This is accomplished by pre-multiplying the known components with a transformation matrix:

$${}^{\mathbf{i+1}}\mathbf{p_i} = {}^{\mathbf{i+1}}\mathbf{X_i}\,{}^{\mathbf{i}}\mathbf{p_i}, \tag{2.5}$$

where ${}^{\mathbf{i}}\mathbf{p_i}$ is a vector associated with link i that is expressed with respect to the i-th coordinate system, ${}^{\mathbf{i+1}}\mathbf{p_i}$ is the same vector expressed in the $(i+1)$-th coordinate system, and ${}^{\mathbf{i+1}}\mathbf{X_i}$ is the 6×6 spatial transformation matrix. This matrix is a function only of the link and joint parameters, and can be defined as:

$${}^{\mathbf{i+1}}\mathbf{X_i} = \begin{bmatrix} {}^{i+1}\mathbf{A}_i & \mathbf{0} \\ {}^{i+1}\mathbf{A}_i\,\tilde{\mathbf{b}}_{i+1}^T & {}^{i+1}\mathbf{A}_i \end{bmatrix}. \tag{2.6}$$

${}^{\mathbf{i+1}}\mathbf{A_i}$ is the 3×3 rotation transformation between the two coordinate systems, and $\mathbf{b_{i+1}}$ is the 3×1 position vector from the origin of the ith coordinate frame to the origin of the origin of frame $(i+1)$, with components expressed in frame i. $\tilde{\mathbf{b}}_{\mathbf{i+1}}$ is the skew-symmetric tensor dual of the vector $\mathbf{b_{i+1}}$, with its component matrix defined by:

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \tag{2.7}$$

$$\tilde{\mathrm{c}} = \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}. \tag{2.8}$$

2.4 Joint model

As stated previously, a *joint* represents the physical interaction between two rigid bodies in a robotic structure.

When considering the simulation of closed chains, a general joint model that allows up to 6 degrees of freedom is very useful. Such a model is described by Lilly [2, p. 14-18] and was also used by Roberson and Schwertassek [17], and Brandl, Johanni and Otter [18]. It can model any constraint and even allows the free motion of two bodies in space to be seen as two bodies that are connected by a 6 degree of freedom joint.

The location of link coordinate frames are the same as described previously when single axis joints are considered. The generalized minimal coordinates that describe the relative motion of body i with respect to body $(i-1)$, are the n_i degrees of freedom of joint i. The joint position vector for joint i is now defined as the $n_i \times 1$ vector q_i^*. For a simple revolute joint q_i^* is just the scalar rotation about the joint axis, and for a prismatic joint it is the scalar displacement along the joint axis.

Now the relative velocity of body i with respect to body $(i-1)$ is a linear function of $\dot{\mathrm{q}}_\mathrm{i}^*$. Likewise the spatial relative acceleration of body i with respect to body $(i-1)$ can be expressed as:

$$\mathrm{a}_\mathrm{i}^\mathrm{r} = \phi_i \ddot{\mathrm{q}}_\mathrm{i}^* + \xi_i \tag{2.9}$$

where ξ_i is a function of joint position, velocity and time (Coriolis, centripetal, gravity and other effects which arise with the use of non-inertial frames).

The $6 \times n_i$ matrix ϕ_i represents the free modes of joint i. Its columns make up a basis for this free vector space, also referred to as the *motion space* of joint i. Because ϕ_i is resolved in the ith coordinate frame, it is frequently constant[3]. A 6×6 matrix can be constructed as:

$$\begin{bmatrix} \phi_1 & \phi_i^c \end{bmatrix} \tag{2.10}$$

where the six columns form a basis for R^6. The columns of ϕ_i^c form a basis for the constrained modes of joint i and is therefore linearly independent of the columns of

[3] It is always constant for joints with less than two rotational degrees of freedom. For physical joints with two or three rotational degrees of freedom, it is a function of the joint rotations.

ϕ_i. A second dual basis may be constructed as follows:

$$[\psi_i \; \psi_i^c]^T = [\phi_i \; \phi_i^c]^{-1} \tag{2.11}$$

From the above definition we have the following relationships:

$$\begin{aligned}
(\phi_i)^T \psi_i &= \mathbf{1} && (2.12)\\
(\phi_i)^T \psi_i^c &= \mathbf{0} && (2.13)\\
(\phi_i^c)^T \psi_i &= \mathbf{0} && (2.14)\\
(\phi_i^c)^T \psi_i^c &= \mathbf{1} && (2.15)
\end{aligned}$$

where **1** and **0** are the identity and zero matrices of the appropriate dimensions. Note from the above that ψ_i^c is a $6 \times (6 - n_i)$ matrix which is orthogonal to ϕ_i. Therefore ψ_i^c represents the constrained modes of joint i, and its columns form a basis for this vector space – the *constraint space* of joint i. Similarly, it can be seen that ψ_i represents the free modes of joint i.

Bodies i and $(i-1)$ can interact through a joint or contact. This interaction can be described by a resultant force and moment: a generalized force, $\mathbf{f_i}$, is exerted on body i by body $(i-1)$ and body i exerts the reaction of equal magnitude but opposite direction on body $(i-1)$. This force vector can be resolved into *components in the motion space and constraint space.* This can be done using either the bases of eq. 2.10 or their dual bases. Using the dual bases, one gets:

$$\mathrm{f_i} = \psi_i \tau_i^* + \psi_i^c \tau_i^c, \tag{2.16}$$

where τ_i^* is the $n_i \times 1$ vector of applied *driving forces*, and τ_i^c is the $(6-n_i) \times 1$ vector of *constraint forces* of joint i. When solving the Direct Dynamics problem, the driving forces are known. The unknowns to be found are the joint accelerations, $\ddot{\mathrm{q}}$, and the constraint forces, τ^c (if desired).

2.5 Kinematic notations

2.5.1 Traditional kinematic notation

Several ways of describing the structure of manipulators have been developed. The most common approach was developed by Denavit and Hartenberg [19]. It makes use of four parameters, and basically describes the angle of twist between successive joints, the rotation around one axis (the z-axis) or displacement along one axis (the z-axis), the perpendicular distance between joint axis. Implicitly it is assumed that only one rotation (for rotational joints) or one displacement (for prismatic joints) is possible. The z-axis is always aligned with the axis of the joint, as shown in Figure 2.2. The parameters are as follows:

a_i = the perpendicular distance along $\hat{\mathrm{x}}_{\mathrm{i}-1}$ between $\hat{\mathrm{z}}_{\mathrm{i}-1}$ and $\hat{\mathrm{z}}_{\mathrm{i}}$.
d_i = the perpendicular distance along $\hat{\mathrm{z}}_{\mathrm{i}}$ between $\hat{\mathrm{x}}_{\mathrm{i}-1}$ and $\hat{\mathrm{x}}_{\mathrm{i}}$.
α_i = the angle about $\hat{\mathrm{x}}_{\mathrm{i}-1}$ between $\hat{\mathrm{z}}_{\mathrm{i}-1}$ and $\hat{\mathrm{z}}_{\mathrm{i}}$.
θ_i = the angle about $\hat{\mathrm{z}}_{\mathrm{i}}$ between $\hat{\mathrm{x}}_{\mathrm{i}-1}$ and $\hat{\mathrm{x}}_{\mathrm{i}}$.

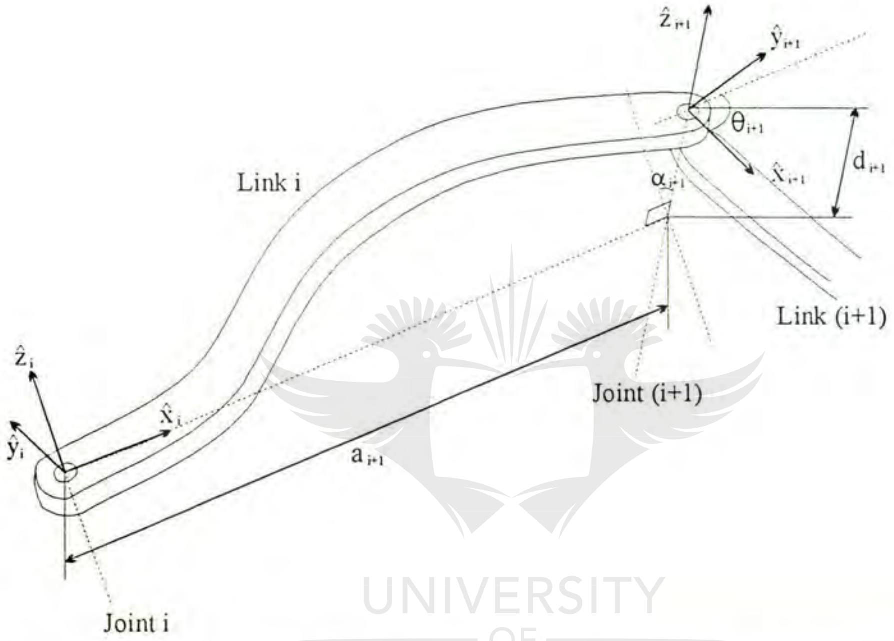


Figure 2.2: Assignment of coordinate frames and link parameters.

From the above it should be clear that this notation is not suitable for use with joints with more than one degree-of-freedom. As all the algorithms in the rest of this work will be aimed at being general in the sense that complex joints with multiple-degrees-of-freedom can be used, a different way of describing the structure of manipulators is needed.

2.5.2 Six degree-of-freedom (SDOF) kinematic notation

We are seeking a kinematic notation from which the relative positions and orientations of the body fixed coordinate frames of a manipulator can be found. It would be very helpful if these parameters can easily be found from the displacements of the joints. These joints must allow up to three rotational degrees of freedom and up to three linear displacement degrees of freedom. A further requirement is that the relative velocities and accelerations of the body fixed coordinate frames must be easily determined from the joint velocities and accelerations. In many robotic systems, the joints are not actuated ball joints (it is hard to imagine such a joint). Many joints have only one degree of freedom. Those joints that do have two or more rotational degrees of freedom, usually consist of two perpendicular joint axes. The two joint axes usually has independent actuators. These two joint axes can either intersect or not. If they do not intersect, this is actually two joints connected by a link of finite length and has to be treated as such. In the case that the axes do intersect, the joint cannot be seen as two joints connected by a mass-less link of zero length for dynamic analyses. This leads to singularities in the inertia matrix for the system. The notation described in this section was specifically developed to handle joints with many intersecting axes and to fulfill the above requirements. The notation was developed with the integration of it with the general joint model and spatial notation in mind.

This notation can be very easily applied to many physical systems, and allows up to six degrees of freedom. It also models real joints more accurately in the sense that the rotation axis for joints with more than two rotational degree of freedom does not stay perpendicular. This notation can easily be extended to include joints where the joint axes are not perpendicular, but it will not be done here since many practical systems with multiple degree of freedom joints have perpendicular joint axes. A very important conceptual similarity of this notation with the previous Denavit-Hartenberg mutant is that the coordinate frames are fixed to the near end of the links.

The notation is based on seeing each joint and link together as a gimbal structure with a three degree-of-freedom prismatic joint connected to the inner ring of the gimbal. The prismatic joint determines where on the link the rotational joint and thus the previous link is attached. The coordinate frame is fixed to the link itself at the intersection of the six joint axis. It has the same orientation as the inner frame of the gimbal. A schematic representation of how a joint is seen is illustrated in Figure 2.3. The rotational degrees of freedom of the joint is now defined as a sequence of three rotations around the three axes of the gimbal:

1. Rotation about the x-axis ($Rot(x)$).

2. Rotation about the transformed y-axis ($Rot(y)$).

3. Rotation about the twice transformed z-axis ($Rot(z)$).

The rotation of the outer frame is the first rotation, the rotation of the frame inside that represent the second rotation. and the rotation of the inner frame is the third rotation. A prismatic joint with three perpendicular axis is fixed to the inner frame of the gimbal. The three axis of this prismatic joint is assumed to be aligned

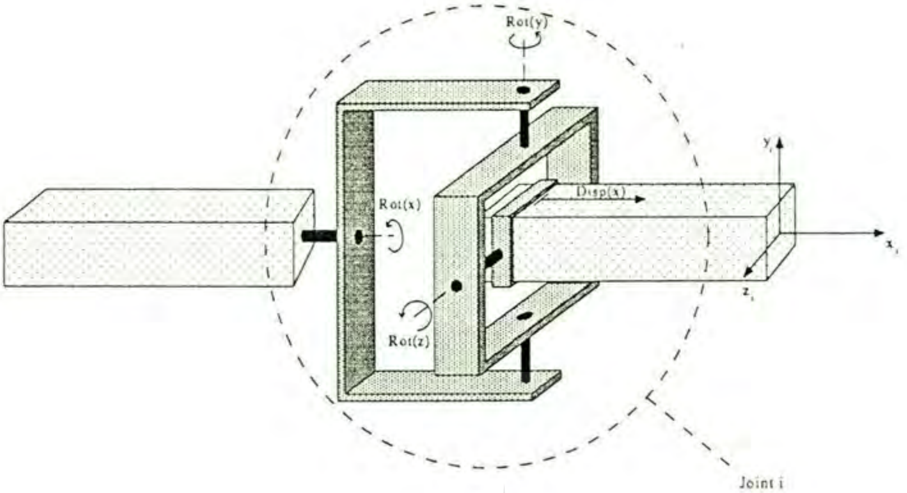


Figure 2.3: The kinematic notation used to describe the structure and geometry of links and joints of structures with multiple degree-of-freedom joints (SDOF kinematic notation). The displacements in the y- and z-directions are not shown.

with the axis of the link coordinate frame[4], which is determined by the orientation of the inner frame of the gimbal. The link coordinate frame is fixed to the near end of the prismatic joint, and has the same orientation as a frame that was fixed to the inner frame of the gimbal in the default position and has undergone the three rotation tranformations. The displacements of this prismatic joint is simply the position vector of the position of the next link frame's origin relative to the previous link's frame origin. The six parameters describing the joint and link geometry of the i-th link is arranged in a column matrix as:

$$\gamma_i = \begin{bmatrix} Rot(x)_i \\ Rot(y)_i \\ Rot(z)_i \\ Disp(x)_i \\ Disp(y)_i \\ Disp(z)_i \end{bmatrix} \tag{2.17}$$

The i-th coordinate frame thus has the same velocity as the i-th link. The i-th link can move relative to the imaginary gimbal frame due to the displacements of the prismatic joint. That implies that the link frame can move relative to the gimbal. In order to simplify the dynamic algorithms, we use the convention that the link frame

[4] This does not have to be the case and this notation can be used without modification when the prismatic axes are not aligned with the link frame axis. Naturally the motion space for the joint would be different.

has the same velocity and acceleration as the link, but at each simulation time step it is repositioned at the intersection of the joint axes of the link. In the next chapter it would soon be clear that the use of this notation together with the general joint model greatly simplifies the dynamic analysis and its algorithmic implementation.

2.6 Rotation matrices and displacement vectors in terms of the link parameters

2.6.1 Denavit-Hartenberg parameters

When using the modified Denavit-Hartenberg notation, the rotation transformation from the frame fixed to the i-th link to the frame fixed to the $(i+1)$-th link, is simply [2]:

$$ {}^{i+1}\mathbf{A}_i = \begin{bmatrix} \cos\theta_{i+1} & \sin\theta_{i+1}\cos\alpha_{i+1} & \sin\theta_{i+1}\sin\alpha_{i+1} \\ -\sin\theta_{i+1} & \cos\theta_{i+1}\cos\alpha_{i+1} & \cos\theta_{i+1}\sin\alpha_{i+1} \\ 0 & -\sin\alpha_{i+1} & \cos\alpha_{i+1} \end{bmatrix}. \tag{2.18}$$

Further, the position of the origin of the $(i+1)$-th coordinate frame relative to the i-th frame is given by:

$$\mathbf{b}_{i+1} = \begin{bmatrix} a_{i+1} \\ -d_{i+1}\sin\alpha_{i+1} \\ d_{i+1}\cos\alpha_{i+1} \end{bmatrix}. \tag{2.19}$$

2.6.2 Six dof kinematic parameters

The first three parameters of this notation describes a sequence of three *simple rotations* that accomplishes the rotation between successive link frames. The rotation transformations for rotations by an angle of θ around the x, y and z-axes respectively is given by:

$$\mathbf{Rot}(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}, \tag{2.20}$$

$$\mathbf{Rot}(y,\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.21}$$

$$\mathbf{Rot}(z,\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.22}$$

Let

$$
\begin{aligned}
\theta_x &= Rot(x)_{i+1}, && (2.23)\\
\theta_y &= Rot(y)_{i+1}, && (2.24)\\
\theta_z &= Rot(z)_{i+1}. && (2.25)
\end{aligned}
$$

Because of the structure of the joint used to describe the first three parameters of the six degree of freedom kinematic notation, the rotations are performed in the order $x - y - z$. The combined transformation is then:

$$
\begin{aligned}
{}^{i+1}\mathbf{A}_i &= \mathbf{Rot}(z, \theta_z)\,\mathbf{Rot}(y, \theta_y)\,\mathbf{Rot}(x, \theta_x) \qquad (2.26)\\
&= \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \qquad (2.27)\\
&= \begin{bmatrix} c\theta_z c\theta_y & c\theta_z s\theta_y s\theta_x + s\theta_z c\theta_x & -c\theta_z s\theta_y c\theta_x + s\theta_z s\theta_x \\ -s\theta_z c\theta_y & -s\theta_z s\theta_y s\theta_x + c\theta_z c\theta_x & z\theta_z c\theta_y c\theta_x + c\theta_z s\theta_x \\ s\theta_y & -c\theta_y s\theta_x & c\theta_y c\theta_x \end{bmatrix} \qquad (2.28)
\end{aligned}
$$

The rotation transformation is thus quite simple to calculate, but because of the author's lack of confidence in his symbolic matrix multiplication abilities, and the difficulty of coding equation 2.28 in a computer algorithm, the rotation transformation is rather calculated using equation 2.27 in the actual algorithms. The position of the origin of the $(i+1)$-th coordinate frame relative to the i-th frame is simply given by the last three components of the link parameters of joint i:

$$
\mathbf{b}_{i+1} = \begin{bmatrix} Disp(x)_i \\ Disp(y)_i \\ Disp(z)_i \end{bmatrix}. \qquad (2.29)
$$

2.6.3 Illustrative example of the application of the six degree of freedom notation

In order to clarify exactly how to use the six degree of freedom kinematic notation, and to illustrate some of the advantages of using this notation, its application to a "Stanford" type manipulator will now be demonstrated. The manipulator is sketched in Figure 2.4. The manipulator is fairly simple: it has rotary and prismatic joints, some joints with two degrees of freedom, and has six degrees of freedom in total. For simplicity, it is assumed that the base coordinate frame (frame 0) coincides with the first frame.

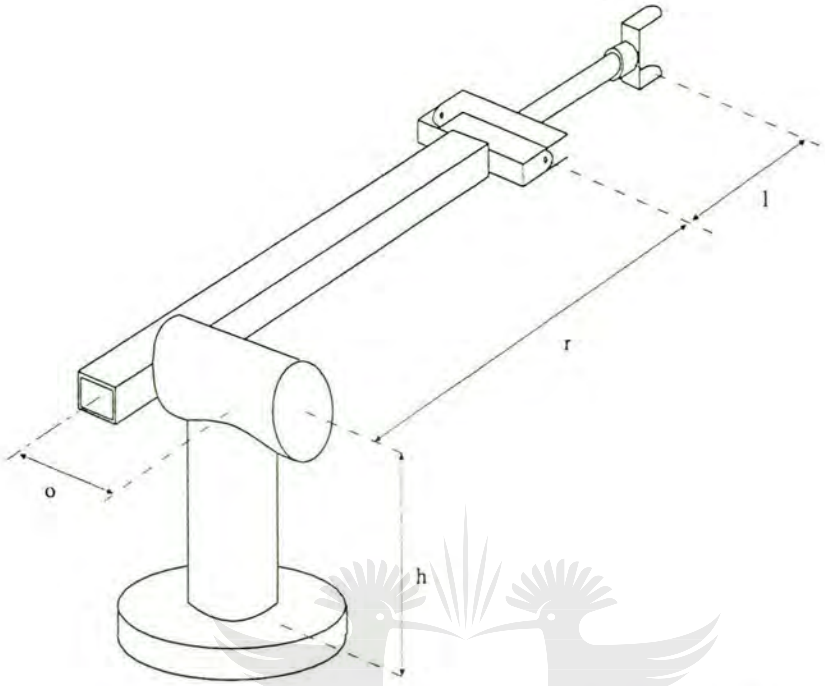


Figure 2.4: The anatomy of a "Stanford manipulator".

When using the SDOF kinematic notation, one can think of it in the following way: you have the necessary number of joint-link units, connected in series, and with all their parameters set to zero. How can the parameters be changed to deform this serial chain into the given system? In general, there will be many parameter combinations which would satisfy this, and this is one of the advantages of using this notation. One has some freedom in choosing how to model the system. When this approach is used, one will always be able to model the system with this notation. One can also use a more traditional approach for determining a set of joint parameters, as in the next paragraph.

No matter what kinematic notation is used, the first step in determining the kinematic parameters is to assign coordinate frames to the various links. The orientation of the respective coordinate frames can be chosen as any convenient orientation. In this example, it looks convenient to choose the coordinate frames as in Figure 2.5. Notice how easy and convenient this choice is. If the Denavit-Hartenberg notation was used, one would have to spend some time to determine how to align the coordinate frame axes: the z-axis has to coincide with the joint axis, the x-axis has to point to the next frame etc. One would also need two more coordinate frames for the system since the D-H notation can handle only one degree of freedom per joint.

Modeling the system using fewer links has many advantages: it is not only simpler, but can result in computational savings in the dynamics algorithms, where the number of calculations needed to determine some quantities are super-linear functions of the number of links in the chain.

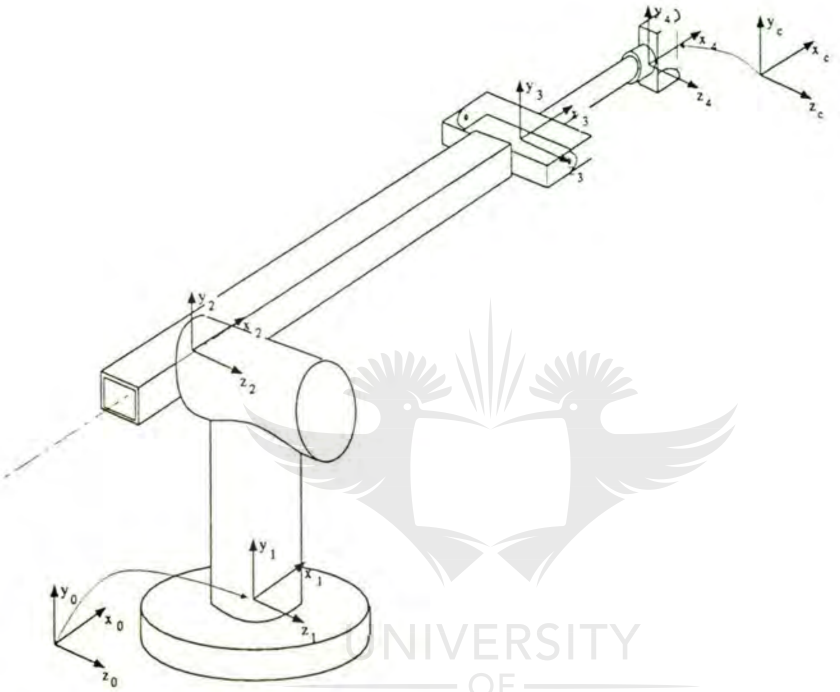


Figure 2.5: The assignment of coordinate frames to the manipulator (not a unique choice).

Determining the joint parameters are now almost trivial. Nonetheless, it will be discussed. Consider the first joint. Since coordinate frame 1 has the same orientation as the base frame, the three rotations of the imaginary gimbal structure are all zero. The displacements of the imaginary prismatic joint joining joint-link 1 to joint-link 2 is simply the components of the relative position of the origin of frame 2 in frame 1: $0\hat{\mathbf{x}}_1 + h\hat{\mathbf{y}}_1 - o\hat{\mathbf{z}}_1$. Joint 1 has only one degree of freedom: rotation about the y-axis. The joint-link parameters and joint space for link 1 is thus:

$$\gamma_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ h \\ -o \end{bmatrix} \tag{2.30}$$

$$\phi_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot (2.31)$$

The second joint-link structure is slightly more interesting. It has a rotary and prismatic joint with intersecting axes. The rotations can take place around the z-axis, while the displacement of the prismatic joint takes place along the x-axis. Once again the three imaginary rotations are all zero for this position. The displacements of the imaginary three degree of freedom prismatic joint is simply $r\hat{\mathbf{x}}_2$. The parameters and joint space for link 2 is:

$$\gamma_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ r \\ 0 \\ 0 \end{bmatrix} \quad (2.32)$$

$$\phi_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.33)$$

The third joint also has two degrees of freedom, but this time both are rotational. This joint looks like the gimbal structure of Figure 2.3, which support the previous claim that many real systems can be modeled by such joints. Coincidentaly, joint 2 uses exactly the first two rotations of the more general gimbal joint. The joint-link parameters and joint space follows:

$$\gamma_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ l \\ 0 \\ 0 \end{bmatrix} \quad (2.34)$$

$$\varphi_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.35}$$

The last joint in the structure is a simple one degree of freedom joint characterized by:

$$\gamma_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m \\ 0 \\ 0 \end{bmatrix} \tag{2.36}$$

$$\phi_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.37}$$

Notice that an extra coordinate frame was fixed to the tip of the chain. This frame is placed where the interaction with the environment is expected to take place. One might reason that this position is already specified by the joint parameters of joint 4. It is indeed so, but the frame is fixed there in any case since a convenient choice for its orientation can be made, which simplifies the description of the interaction with the environment (directions of constraint etc.).

2.6.4 Other considerations

With any joint that has more than one rotational degree of freedom (excluding ball-and-socket joints), the joint space of the joint (expressed in the coordinate frame of the particular joint) is not a constant base, but it is a function of the joint parameters (specifically the rotations). Only the upper three rows of the joint space is affected because the displacements of the prismatic joints has no effect on the orientation of any coordinate frame in the system.

The above implies that a transformed joint space has to be calculated for each joint position where the current joint space is needed. Fortunately, this is not difficult with the kinematic notation we are using. Given the joint space in the default position of the joint (the first three joint parameters all zero), the joint space corresponding to any set of joint parameters can be found as follows: the direction of the third rotation axis of joint i is not affected by the three rotations of joint i, since the last rotation corresponds to a simple transformation. The direction of the second axis of joint i is only affected by the third rotation and simply transforms to:

$$\mathbf{Rot}(z, \theta_z) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{2.38}$$

which is simply the second column of $\mathbf{Rot}(z, \theta_z)$. The direction of the first axis in frame i is affected by both the second and third rotation:

$$\mathbf{Rot}(z, \theta_z)\,\mathbf{Rot}(y, \theta_y) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{2.39}$$

which effectively involves only one rotation transformation, being the rotation around z of the first column of the rotation transformation around y. All these rotation transformations are known. The calculation of the transformed joint space therefore uses at most nine multiplications and 14 additions per joint.

2.7 Advantages of the SDOF kinematic notation

Some of the advantages of the six degree of freedom kinematic notation should now be apparent:

1. It can describe joints with up to six degrees of freedom.
2. Has an easy physical interpretation.
3. Is easy to apply to most physical systems.
4. Fewer joints and links in systems with joints with more than one d.o.f. than if Denavit-Hartenberg notation were used:
   (a) Conceptually simpler.
   (b) Computational savings in dynamics algorithms.
5. If Denavit-Hartenberg notation were applied to systems with multiple degree of freedom joints, massless links would have to be created in the system, to accommodate more degrees of freedom at a joint without affecting the inertial properties of the structure. This can lead to singularities in the inertia matrix of the system (joint space inertia matrix).

6. There is a direct relationship between the joint positions and the kinematic parameters, which greatly simplifies the calculation of the rotation transformations and relative position vectors for different joint positions.

Chapter 3

Cartesian Tensors in dynamics of rigid-body systems

In Chapter 1 it has been shown that rotations can be represented by the rotation tensor. Here it will be shown that tensors can be used extensively in the dynamic equations for rigid bodies, and that they develop quite naturally from the dynamic analysis.

3.1 Angular velocity tensor

The orientation of a body in three dimensional Euclidean space as a function of time is described by the rotation tensor, $\mathbf{R} \equiv \mathbf{R}(t)$. The kinematic analysis of the pure rotational motion of a rigid body can be analyzed by only considering $\mathbf{R}$ and it's first and second time derivatives. To introduce the angular velocity tensor, we consider the following theorem concerning the time derivative of orthogonal tensors:

Theorem 1 *Any differentiable orthogonal tensor* $\mathbf{Q} \equiv \mathbf{Q}(t)$ *satisfies the following first order differential equation:*

$$\dot{\mathbf{Q}} = \mathbf{\Phi}\mathbf{Q} \tag{3.1}$$

where $\mathbf{\Phi}$ *is a second order skew-symmetric tensor.*

The proof of this can be found in [9]. Since $\mathbf{R}$ is an orthogonal tensor, as a corollary to the above theorem, $\mathbf{R}$ satisfies the differential equation

$$\dot{\mathbf{R}} = \mathbf{\Phi}\mathbf{R} \tag{3.2}$$

where $\mathbf{\Phi}$ is still a second order skew-symmetric tensor. When the rotation tensor, $\mathbf{R}$, is defined in a 3-D Euclidean space, it describes the orientation of a rigid body, and it's first time derivative describes the time rate of change of it's orientation. We now call the skew-symmetric tensor $\mathbf{\Phi}$ that relates the time rate of change of orientation to the orientation, the angular velocity tensor, and denote it by $\tilde{\omega}$:

$$\dot{\mathrm{R}} = \tilde{\omega}\mathrm{R} \tag{3.3}$$

Equation 3.3 can be rewritten as

$$\tilde{\omega} = \dot{\mathrm{R}}\mathrm{R}^T \tag{3.4}$$

which can be used as the definition of the angular velocity tensor. The fact that $\mathbf{R}$ is orthogonal can be used to ensure that $\mathbf{R}^{-1} = \mathbf{R}^T$. Because $\tilde{\omega}$ is defined in a 3-D Euclidean space and it is skew-symmetric, it has a unique dual vector, ω, which is the angular velocity vector of classic vector mechanics. A complete treatment of vectors and their duals can be found in [9].

3.2 Angular acceleration tensor

In order to arrive at a definition of the angular acceleration tensor, we seek the functional relationship between the second time derivative of the orientation relative to an inertial frame, $\ddot{\mathrm{R}}$, and $\mathbf{R}$. It follows from the following theorem:

Theorem 2 *Any differentiable orthogonal tensor* $\mathbf{Q} \equiv \mathbf{Q}(t)$ *satisfies the following first order differential equation:*

$$\ddot{\mathrm{Q}} = \mathbf{\Psi}\mathrm{Q} \tag{3.5}$$

where $\mathbf{\Psi}$ *is the second order tensor defined by*

$$\mathbf{\Psi} = \dot{\mathbf{\Phi}} + \mathbf{\Phi}^2 \tag{3.6}$$

with $\mathbf{\Phi}$*, the angular velocity tensor, defined by*

$$\mathbf{\Phi} = \dot{\mathrm{Q}}\mathrm{Q}^T \tag{3.7}$$

and the square of $\mathbf{\Phi}$ *is the dot product of* $\mathbf{\Phi}$ *with itself.*

As a corollary to this theorem, the second order rotation tensor $\mathbf{R}$ satisfies the differential equation

$$\ddot{\mathrm{R}} = \mathbf{\Psi}\mathrm{R} \tag{3.8}$$

where $\mathbf{\Psi}$ is the second order tensor defined by equation 3.6. This is valid for orthogonal vectors in n-D Euclidean space. From equation 3.8 we have the functional relationship between the zeroth and the second time derivative of the orientation tensor. For the particular case of 3-D Euclidean space, the following notation can be introduced:

$$\ddot{\mathrm{R}} = \mathcal{A}\mathrm{R}. \tag{3.9}$$

By using the angular velocity for 3-D Euclidean space, $\tilde{\omega}$, equation 3.6 is written as:

$$\mathcal{A} = \dot{\tilde{\omega}} + \tilde{\omega}\tilde{\omega} \tag{3.10}$$

which defines the angular acceleration tensor $\mathcal{A}$. Equation 3.9 can also be used to define the angular acceleration tensor:

$$\mathcal{A} = \ddot{\mathrm{R}}\mathrm{R}^T \tag{3.11}$$

$\dot{\mathbf{\Phi}}$ is skew-symmetric and $\mathbf{\Phi}^2$ is symmetric, from which it follows that $\mathbf{\Psi}$ (and also the special case $\mathcal{A}$) is neither symmetric nor skew-symmetric, since it is the sum of the above. In the 3-D case, since the tensor $\tilde{\omega}\tilde{\omega}$ is symmetric, the vector invariant of the angular acceleration tensor $\mathcal{A}$ is simply the vector invariant of $\dot{\tilde{\omega}}$, which is the angular acceleration vector. From the above, it follows that there does not exist a one-to-one relationship between the angular acceleration vector and tensor.

3.3 Linear velocity and acceleration in terms of angular tensors

To arrive at the relationship between the first and second time derivatives of vectors in general, we consider the special case of the position vector. Consider two coordinate systems – an inertial coordinate system and a body coordinate system which is denoted by $\{\mathrm{e}\}$ and $\{\mathrm{e}'\}$ respectively (Figure 3.1). As always, the body frame is fixed to the rigid body and moves with it. Consider a point on the rigid body, denoted by $\mathbf{p}$. Let the position vector of $\mathbf{p}$ be $\mathbf{r_1}$ relative to the origin, $\mathbf{o}$, of the inertial frame, and $\mathbf{r_2}$ relative to the origin, $\mathbf{o}'$, of the body fixed frame. Let $\mathbf{s}$ be the vector from $\mathbf{o}$ to $\mathbf{o}'$. The vectors are related by

$$\mathbf{r_1} = \mathbf{s} + \mathbf{r_2} \tag{3.12}$$

The orientation of the moving coordinate system relative to the inertial system is specified by the rotation tensor $\mathbf{R}$. Let $\mathbf{r}_2'$ denote the position vector of point $\mathbf{p}$ relative to $\mathbf{o}'$, expressed in $\{\mathrm{e}'\}$. Then

$$\mathbf{r_2} = \mathbf{R}\mathbf{r}_2' \tag{3.13}$$

and equation 3.12 can also be written as:

$$\mathbf{r_1} = \mathbf{s} + \mathbf{R}\mathbf{r}_2' \tag{3.14}$$

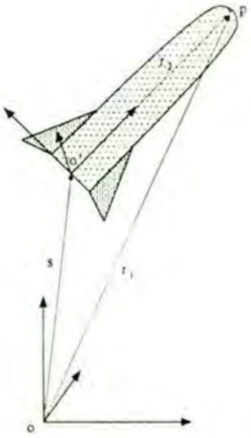


Figure 3.1: The use of inertial and body fixed coordinate frames.

The absolute linear velocity of point $\mathbf{p}$ is defined as the first time derivative of its position vector relative to the inertial coordinate system. This can be computed with equation 3.14:

$$\dot{\mathbf{r}}_1 = \dot{\mathbf{s}} + \dot{\mathbf{R}}\mathbf{r}_2' \tag{3.15}$$

where the time derivative of $\mathbf{R}\mathbf{r}_2'$ is simply $\dot{\mathbf{R}}\mathbf{r}_2'$ because $\dot{\mathbf{r}}_2' = \mathbf{0}$ because $\mathbf{r}_2'$ is constant in the body coordinate system when the body is rigid. Substituting $\dot{\mathbf{R}}$ from equation 3.3, and using equation 3.13, we arrive at an elegant expression for the linear velocity:

$$\dot{\mathbf{r}}_1 = \dot{\mathbf{s}} + \tilde{\omega}\mathbf{r}_2 \tag{3.16}$$

In a similar way, the second time derivative of equation 3.14 defines the vector of absolute linear acceleration of point p. With the use of equation 3.9 it can be simplified as:

$$\ddot{\mathbf{r}}_1 = \ddot{\mathbf{s}} + \mathcal{A}\mathbf{r}_2 \tag{3.17}$$

The time derivatives of vectors that are constant relative to the body coordinate frame is written as:

$$\dot{\mathbf{r}}_2 = \tilde{\omega}\mathbf{r}_2 \tag{3.18}$$

and

$$\ddot{\mathbf{r}}_2 = \mathcal{A}\mathbf{r}_2 \tag{3.19}$$

from equation 3.13 and equations 3.3 and 3.9 respectively. The velocity and acceleration vectors as expressed above can easily be seen to be equivalent to their expressions in terms of vectors, where $\tilde{\omega}$ is equivalent to $\omega\times$ (the cross product of ω with whatever follows).

3.4 Rigid body inertia tensor

The rigid body inertia tensor emerges naturally when the angular momentum or kinetic energy of a rigid body is evaluated [3]. From these analyses, the inertia tensor around a point $\mathbf{o}$ is defined as:

$$\mathbf{I}_o = \int_m (\mathbf{r} \cdot \mathbf{r}\mathbf{1} - \mathbf{r} \otimes \mathbf{r}) dm \tag{3.20}$$

where $\mathbf{r}$ is the position vector of the point dm from the point $\mathbf{o}$, and $\mathbf{r} \otimes \mathbf{r}$ is the tensor product of $\mathbf{r}$ with itself. The above is used in the vector treatment of Newtonian dynamic analysis. For the Cartesian tensor analysis of dynamics, a different definition of is used. The Euler tensor is defined in [1] as:

$$\mathbf{J}_o = \int_m \mathbf{r} \otimes \mathbf{r} dm \tag{3.21}$$

It will later be seen that the Euler tensor leads to more efficient expressions in tensor dynamic analysis than the inertia tensor. The inertia and Euler tensor both describe the same property of a rigid body and they should therefore be equivalent. The equivalence can be proven as follows (as in [9]): equation 3.20 can be written as:

$$\mathbf{I}_o = \int_m \tilde{\mathbf{r}}\tilde{\mathbf{r}} dm \tag{3.22}$$

The tensor product of two vectors can be written in terms of their dot product and the dot product of their dual tensors as:

$$\mathbf{v} \otimes \mathbf{u} = \tilde{\mathbf{u}}\tilde{\mathbf{v}} + (\mathbf{v} \cdot \mathbf{u})\mathbf{1} \tag{3.23}$$

The dot product of two vectors can be found from the dot product of their duals as follows:

$$\mathbf{u} \cdot \mathbf{v} = -\frac{1}{2} tr[\tilde{\mathbf{u}}\tilde{\mathbf{v}}] \tag{3.24}$$

With the help of the above two equations, equation 3.21 can be manipulated as follows:

$$\begin{aligned}
\mathbf{J}_o &= \int_m (\tilde{\mathbf{r}}\tilde{\mathbf{r}} + \mathbf{r} \cdot \mathbf{r}\mathbf{1}) dm && (3.25) \\
&= \int_m \tilde{\mathbf{r}}\tilde{\mathbf{r}} dm + \mathbf{1} \int_m -\frac{1}{2} tr[\tilde{\mathbf{r}}\tilde{\mathbf{r}}] dm && (3.26) \\
&= \int_m \tilde{\mathbf{r}}\tilde{\mathbf{r}} dm - \frac{1}{2} tr[\int_m \tilde{\mathbf{r}}\tilde{\mathbf{r}} dm]\mathbf{1} && (3.27) \\
&= -\mathbf{I}_o + \frac{1}{2} tr[\mathbf{I}_o]\mathbf{1} && (3.28)
\end{aligned}$$

The relationship between the inertia and Euler tensor is thus:

$$\mathbf{J}_o = \frac{1}{2} tr[\mathbf{I}_o]\mathbf{1} - \mathbf{I}_o \tag{3.29}$$

and similarly

$$\mathbf{I}_o = tr[\mathbf{J}_o]\mathbf{1} - \mathbf{J}_o \tag{3.30}$$

In practice, the inertia tensor is determined experimentally. This is easiest to do when it is evaluated around the center of mass of the body - yielding $\mathbf{I}_c$. For dynamic analyses the inertia around a point other than the center of mass is required. This can be calculated from $\mathbf{I}_c$ with the *parallel axis theorem:*

$$\mathbf{I}_o = \mathbf{I}_c + m[\mathbf{r}_c \cdot \mathbf{r}_c \mathbf{1} - \mathbf{r}_c \otimes \mathbf{r}_c] \tag{3.31}$$

where $\mathbf{r}_c$ is the position vector of the center of mass relative to point $\mathbf{o}$, and m is the mass of the body. Equation 3.31 can also be written more compactly as:

$$\mathbf{I}_o = \mathbf{I}_c - m\tilde{\mathbf{r}}_c\tilde{\mathbf{r}}_c \tag{3.32}$$

Proof of the parallel axis theorem can be found in [3]. It can be expected that a similar theorem exists for the Euler tensor. Such a theorem indeed exists and proof of it can be found in [9].

Theorem 3 (parallel axis theorem): *When the Euler tensor of a rigid body around its center of mass, $\mathbf{J}_c$, is known, then the Euler tensor about any other point $\mathbf{o}$, $\mathbf{J}_o$, is given by:*

$$\mathbf{J}_o = \mathbf{J}_c + m\mathbf{r}_c \otimes \mathbf{r}_c \tag{3.33}$$

where $\mathbf{r}_c$ is the position vector of the center of mass relative to point $\mathbf{o}$.

This can be proven by using the parallel axis theorem for the rigid body inertia (equation 3.31) and equation 3.29.

The inertia and Euler tensor is described relative to a coordinate frame by a set of components called the moments of inertia and the products of inertia. If these components are described in an inertial frame and the body is moving, these components will be time dependent. The description of these tensors relative to a body fixed frame leads to time independent components - which is advantageous, but also introduces the need to be able to transform the description of these tensors between coordinate frames.

If $\mathbf{I}_c$ is the description relative to an inertial frame and $\mathbf{I}_{c'}$ is the description of the inertia tensor relative to the body fixed frame and $\mathbf{R}$ is the rotation tensor from the inertial frame to the body frame, these descriptions are related by:

$$\mathbf{I}_c = \mathbf{R}\mathbf{I}_{c'}\mathbf{R}^T \tag{3.34}$$

This transformation holds for all tensors. The above equation can be used to write the time derivative of any tensor, and specifically the inertia tensor, in terms of the tensor and the angular velocity tensor:

$$\begin{aligned}
\dot{\mathbf{I}}_c &= \dot{\mathrm{R}}\mathbf{I}_{c'}\mathrm{R}^T + \mathrm{R}\mathbf{I}'_c\dot{\mathrm{R}}^T && (3.35)\\
&= \tilde{\omega}\mathbf{I}_{c'}\mathrm{R}^T + \mathrm{R}\mathbf{I}_{c'}\mathrm{R}^T\tilde{\omega}^T && (3.36)\\
&= \tilde{\omega}\mathbf{I}_c + \mathbf{I}_c\tilde{\omega}^T && (3.37)\\
&= \tilde{\omega}\mathbf{I}_c - \mathbf{I}_c\tilde{\omega} && (3.38)
\end{aligned}$$

where the fact that $\mathbf{I}_{c'}$ is time independent and $\tilde{\omega}$ is skew-symmetric has been used. From equation 3.29 it follows that $\dot{\mathbf{I}}_o = -\dot{\mathbf{J}}_o$ (the trace of a tensor is a scalar invariant).

3.5 Angular momentum tensor

In order to investigate the angular momentum tensor, we first consider the angular momentum vector around the origin, $\mathbf{o}$, of an inertial coordinate system - $\mathbf{L}_o$. Angular momentum is defined in an inertial reference frame. In classical vector mechanics [3], this vector is defined by:

$$\mathbf{L}_o = \mathbf{s} \times [\dot{\mathbf{s}} + \omega \times \mathbf{r}_c]m + \mathbf{r}_c \times \dot{\mathbf{s}}m + \mathbf{I}_{o'}\omega \qquad (3.39)$$

where $\mathbf{o}'$ is a point fixed on the rigid body, $\mathbf{s}$ is the position vector of $\mathbf{o}'$ relative to $\mathbf{o}$, $\mathbf{r}_c$ is the position of the center of mass relative to $\mathbf{o}'$, and $\mathbf{I}_{o'}$ is the rigid body inertia tensor about point $\mathbf{o}'$, and ω is the angular velocity vector of the rigid body. Equation 3.39 reduces if $\mathbf{o}'$ is fixed in space ($\dot{\mathbf{s}} = \mathbf{0}$), or if the center of mass is used as the reference point $\mathbf{o}'$ ($\mathbf{r}_c = \mathbf{0}$). The middle term in equation 3.39 then vanishes and the rest of the equation can easily be interpreted. The first term is the angular momentum due to the translation of the center of mass, while the last term is the angular momentum due to the angular velocity of the body.

To simplify the following analysis, we assume that the body has no translational motion ($\dot{\mathbf{s}}$), and that the inertial coordinate system has its origin at the point $\mathbf{o}'$. Equation 3.39 then becomes:

$$\mathbf{L}_o = \mathbf{I}_o\omega \qquad (3.40)$$

When the center of mass is also the center of rotation, equation 3.40 is written as:

$$\mathbf{L}_c = \mathbf{I}_c\omega \qquad (3.41)$$

If the center of rotation is not the center of mass, it is still useful to write the angular momentum vector in term of the angular momentum around the center of mass, as this is more convenient to calculate:

$$\begin{aligned}
\mathbf{L}_o &= \mathbf{I}_o\omega && (3.42)\\
&= [\mathbf{I}_c - m\tilde{\mathbf{r}}_c\tilde{\mathbf{r}}_c]\omega \qquad \text{(parallel axis theorem)} && (3.43)\\
&= \mathbf{L}_c - m\tilde{\mathbf{r}}_c\tilde{\mathbf{r}}_c\omega && (3.44)\\
&= \mathbf{L}_c + m\tilde{\mathbf{r}}_c\dot{\mathbf{r}}_c && (3.45)
\end{aligned}$$

The angular momentum vector can be written in terms of the Euler tensor by substituting equation 3.30 in equation 3.40:

$$\mathbf{L}_o = -\mathbf{J}_o\omega + \frac{1}{2}tr[\mathbf{J}_o]\omega \tag{3.46}$$

Since there is a one-to-one relationship between a vector and its second order skew-symmetric Cartesian tensor dual, the angular momentum can also be described by a skew-symmetric tensor by simply applying the dual operator on the angular momentum vector:

$$\tilde{\mathbf{L}}_o = dual(\mathbf{L}_o) \tag{3.47}$$

The skew-symmetric tensor $\tilde{\mathbf{L}}_o$ is called the angular momentum tensor about the point $\mathbf{o}$.

Equation 3.47 provides an indirect definition of the angular momentum tensor through the use of the angular momentum vector. It is possible to define the angular momentum tensor directly in terms of the inertia tensor and the angular velocity tensor. To do this, we proceed from equation 3.47 with $\mathbf{L}_o$ defined by equation 3.40:

$$\tilde{\mathbf{L}}_o = dual(\mathbf{I}_o\omega) \tag{3.48}$$

and use the following proposition that is proven in [9]:

Proposition 1 *Let* $\mathbf{S}$ *be a symmetric tensor and* $\mathbf{v}$ *any vector. Then the dual tensor,* $dual(\mathbf{S}\cdot\mathbf{v})$*, satisfies the following equation:*

$$dual(\mathbf{S}\cdot\mathbf{v}) = -[\mathbf{S}\tilde{\mathbf{v}} + \tilde{\mathbf{v}}\mathbf{S}] + tr[\mathbf{S}]\tilde{\mathbf{v}} \tag{3.49}$$

Since the inertia tensor $\mathbf{I}_o$ is symmetric, the angular momentum tensor can be expanded from its dual definition as:

$$\begin{aligned}
\tilde{\mathbf{L}}_o &= -[\mathbf{I}_o\tilde{\omega} + \tilde{\omega}\mathbf{I}_o] + tr[\mathbf{I}_o]\tilde{\omega} && (3.50)\\
&= \mathbf{I}_o\tilde{\omega}^T - \tilde{\omega}\mathbf{I}_o + tr[\mathbf{I}_o]\tilde{\omega} && (3.51)\\
&= [\tilde{\omega}\mathbf{I}_o]^T - \tilde{\omega}\mathbf{I}_o + tr[\mathbf{I}_o]\tilde{\omega} && (3.52)
\end{aligned}$$

which defines the angular momentum tensor directly. Equation 3.52 can also be written in terms of the Euler tensor – which is indeed more compact. If we substitute equation 3.30 into equation 3.52 we get:

$$\begin{aligned}
\tilde{\mathbf{L}}_o &= [tr[\mathbf{J}_o]\tilde{\omega} - \tilde{\omega}\mathbf{J}_o]^T - [tr[\mathbf{J}_o]\tilde{\omega} - \tilde{\omega}\mathbf{J}_o] + 2tr[\mathbf{J}_o]\tilde{\omega} && (3.53)\\
&= -tr[\mathbf{J}_o]\tilde{\omega} + \mathbf{J}_o\tilde{\omega} - tr[\mathbf{J}_o]\tilde{\omega} + \tilde{\omega}\mathbf{J}_o + 2tr[\mathbf{J}_o]\tilde{\omega} && (3.54)\\
&= \mathbf{J}_o\tilde{\omega} + \tilde{\omega}\mathbf{J}_o && (3.55)\\
&= \mathbf{J}_o\tilde{\omega} - [\mathbf{J}_o\tilde{\omega}]^T && (3.56)
\end{aligned}$$

When the rigid body is rotating around its center of mass, the subscripts **o** can simply be replaced by **c**. From the above it can be seen that the angular momentum vector has a simpler expression when expressed in terms of the inertia tensor, while the angular momentum tensor has a simpler expression when the Euler tensor is used.

3.6 Equations of motion

As stated before, the equations of motion establishes the relationships between the motion of a body and the forces and torques acting on it. The motion of a rigid body can be decomposed into two independent motions: a pure translational motion of a point on the body (usually its center of mass), and the rotational motion of the body around that point. The Newton-Euler formalism makes use of this logical decomposition and uses two sets of equations to describe the rigid body motion. The translational motion is described by Newton's second axiom as:

$$\mathbf{F}_c = \frac{d}{dt}(m\dot{\mathbf{r}}_c), \tag{3.57}$$

where the position vector is measured in inertial space. The rotational motion is described by Euler's axiom. Euler's axiom of rotational motion relates the time rate of change of angular momentum to the torque vector (in inertial space):

$$\mathbf{M}_o = \dot{\mathbf{L}}_o \tag{3.58}$$

It is derived from Newton's second axiom, where a body is considered as a finite set of point masses and the distance between these point masses is assumed to be kept constant by internal forces. The most important assumption made is that the internal forces do not cause a resultant moment. These assumptions should be kept in mind when applying Euler's axiom.

Not much is gained by writing Newton's axiom in terms of tensors. One can use the angular acceleration tensor to express the linear acceleration[1] of the point **c** in

[1] In rigid body mechanics, a basic assumption is that the mass of the bodies remain constant, thus Newton's axiom simplifies to $\mathbf{F}_c = m\ddot{\mathbf{r}}_c$.

terms of its position in a body fixed frame and the linear acceleration of the body fixed frame relative to an inertial frame. A tensor formulation of Euler's axiom has more advantages and will be considered.

If a body rotates around it's center of mass, Euler's axiom can be written and expanded as:

$$\begin{aligned}
\mathbf{M}_c &= \dot{\mathbf{L}}_c && (3.59)\\
&= \frac{d}{dt}\left(\mathbf{I}_c\omega\right) && (3.60)\\
&= \dot{\mathbf{I}}_c\omega + \mathbf{I}_c\dot{\omega} && (3.61)
\end{aligned}$$

But it has been shown that the time derivative of the inertia tensor can be expressed in terms of the angular velocity and inertia tensor (equation 3.38):

$$\dot{\mathbf{I}}_c = \tilde{\omega}\mathbf{I}_c - \mathbf{I}_c\tilde{\omega} \qquad (3.62)$$

Therefore

$$\begin{aligned}
\mathbf{M}_c &= \left[\tilde{\omega}\mathbf{I}_c - \mathbf{I}_c\tilde{\omega}\right]\omega + \mathbf{I}_c\dot{\omega} && (3.63)\\
&= \mathbf{I}_c\dot{\omega} + \tilde{\omega}\mathbf{I}_c\omega && (3.64)
\end{aligned}$$

which is similar to the expression found in vector mechanics. If the rotation is not around the center of mass, the following relationship can be used to rewrite equation 3.64:

$$\mathbf{M}_o = \mathbf{M}_c + \mathbf{r}_c \times \mathbf{F}_c \qquad (3.65)$$

where $\mathbf{F_c}$ is the total force caused at the center of mass of the rigid body due to its rotational motion. After some manipulation [9, chapter 4], it can be shown that the vector $\mathbf{M_o}$ satisfies the equation

$$\mathbf{M_o} = \mathbf{I_o}\dot{\omega} + \tilde{\omega}\mathbf{I_o}\omega \qquad (3.66)$$

By substituting equation 3.30 the moment vector can be expressed in terms of the Euler tensor:

$$\mathbf{M}_o = \left[\mathbf{J}_o\dot{\omega} + \tilde{\omega}\mathbf{J}_o\omega\right] + tr[\mathbf{J}_o]\dot{\omega} \qquad (3.67)$$

The torque tensor can now be defined from the torque vector by using the dual operator:

$$\tilde{\mathbf{M}}_o = dual\left(\tilde{\mathbf{M}}_o\right) \qquad (3.68)$$

It would be advantageous to be able to arrive at a definition of the torque tensor without having to resort to vector mechanics. This is indeed possible. It follows from Euler's axiom. The dual operator can be applied to both sides of equation 3.58 to yield:

$$\tilde{\mathbf{M}}_o = \frac{d}{dt}\tilde{\mathbf{L}}_o \tag{3.69}$$

The angular momentum tensor, $\tilde{\mathbf{L}}_o$, has already been defined by using other tensors (equation 3.52), and can be substituted into equation 3.69:

$$\tilde{\mathbf{M}}_o = \left[\dot{\tilde{\omega}}\mathbf{I}_o + \tilde{\omega}\dot{\mathbf{I}}_o\right]^T - \left[\dot{\tilde{\omega}}\mathbf{I}_o + \tilde{\omega}\dot{\mathbf{I}}_o\right] + tr[\mathbf{I}_o]\dot{\tilde{\omega}} \tag{3.70}$$

where the trace of a tensor (and thus $\mathbf{I}_o$) is a scalar invariant, thus not time dependent. Equation 3.38 for rotation about **o** can be substituted for the time derivative of the inertia tensor:

$$\begin{aligned}
\tilde{\mathbf{M}}_o &= \left[\dot{\tilde{\omega}}\mathbf{I}_o + \tilde{\omega}\tilde{\omega}\mathbf{I}_o - \tilde{\omega}\mathbf{I}_o\tilde{\omega}\right]^T - \left[\dot{\tilde{\omega}}\mathbf{I}_o + \tilde{\omega}\tilde{\omega}\mathbf{I}_o - \tilde{\omega}\mathbf{I}_o\tilde{\omega}\right] + tr[\mathbf{I}_o]\dot{\tilde{\omega}} && (3.71)\\
&= \left[(\dot{\tilde{\omega}} + \tilde{\omega}\tilde{\omega})\mathbf{I}_o\right]^T - \left[(\dot{\tilde{\omega}} + \tilde{\omega}\tilde{\omega})\mathbf{I}_o\right] + [\tilde{\omega}\mathbf{I}_o\tilde{\omega}]^T - [\tilde{\omega}\mathbf{I}_o\tilde{\omega}] + tr[\mathbf{I}_o]\dot{\tilde{\omega}} && (3.72)\\
&= \left[(\dot{\tilde{\omega}} + \tilde{\omega}\tilde{\omega})\mathbf{I}_o\right]^T - \left[(\dot{\tilde{\omega}} + \tilde{\omega}\tilde{\omega})\mathbf{I}_o\right] + tr[\mathbf{I}_o]\dot{\tilde{\omega}} && (3.73)\\
&= [\mathcal{A}\mathbf{I}_o]^T - \mathcal{A}\mathbf{I}_o + tr[\mathbf{I}_o]\dot{\tilde{\omega}} && (3.74)
\end{aligned}$$

which defines the torque tensor without using vector mechanics. The torque tensor can be defined more efficiently if the Euler tensor is used instead of the inertia tensor. By following the same principles as above, the result is:

$$\tilde{\mathbf{M}}_o = \mathcal{A}\mathbf{J}_o - [\mathcal{A}\mathbf{J}_o]^T \tag{3.75}$$

Equation 3.75 is the tensor formulation of the generalized Euler equation of rigid body rotational motion.

Chapter 4

Inverse dynamics of manipulators

> *Space, time, and force are a priori forms; they can be derived only from contemplation and from general principles of research. Their common relation to each other in mechanics must be regarded as something inspired indeed by experience but in its generality fixed by convention.*
>
> **Hamel**
> Elementare Mechanik (1912)

4.1 Introduction

The inverse dynamics problem is concerned with determining the joint forces[1] when the joint positions, velocities and accelerations are given. In other words, it is concerned with determining the necessary applied joint forces needed to attain a prescribed manipulator motion. From a kinematic point of view, one can always determine the motion of any link in the manipulator when the joint positions, velocities and accelerations are given[2]. The inverse is unfortunately not true, that is the joint velocities and accelerations of all links cannot always be determined in a unique way if the position, velocity and acceleration of the end of the manipulator is given[3]. Since it seems so logical to determine the link motions from the joint motions, one can just as well take the next step and apply Newton's second axiom to each link and solve for the forces that should be acting on the link. One can of course use any dynamic formalism of your choice to solve the inverse dynamics problem, and

[1] When referring to forces, it will usually include the linear forces and rotational moments, especially if used in any context where the spatial notation and general joint model is used.

[2] This is called the forward kinematic problem

[3] Notice that we talk as if any manipulator or structure has an end. This is because any multichain manipulator can be broken into separate serial chains and these serial chains considered for forward kinematic and inverse dynamic analyses. To some extent, this is also true for the forward dynamic analyses.

one is guaranteed to be able to arrive at similar algorithms with exactly the same computational costs [8, 9]. Because of the direct physical insight when using the Newton-Euler formalism, and the easy integration of the general joint model, spatial notation and SDOF kinematic notation with this formalism, the Newton-Euler formalism is preferred by the author and is the one that will be used in this chapter.

It was mentioned before that the structure of the algorithms and the representation of quantities has a direct influence on the efficiency of the algorithms. A recursive algorithm will be derived. A forward recursion (from the base to the tip) will be used to calculate the accelerations of the links in the manipulator. Newton and Eulers' axioms will be used to calculate the forces acting on each link with a backward recursion (from the tip of the chain to the base). From the resultant force that acts on each link, the joint forces can easily be determined during the backward recursion. The accelerations and dynamic quantities will be represented in the respective link coordinate frames, and the equations of motion will also be expressed in these link coordinate frames [10]. Spatial notation [20],with the slight modification as discussed before, will be used. The resulting algorithm should be very similar to an algorithm proposed by Featherstone [21].

4.2 Spatial inertia

The components of the spatial inertia of a body is expressed as a 6 × 6 matrix. It is expressed in the local link coordinate frame and is defined from the mass, inertia tensor and position of the center of gravity as:

$$\mathbf{I_i} = \begin{bmatrix} \bar{\mathbf{I}}_i & \tilde{\mathbf{h}}_\mathbf{i} \\ \tilde{\mathbf{h}}_\mathbf{i}^\mathbf{T} & \mathbf{M_i}, \end{bmatrix} \tag{4.1}$$

where $\mathbf{M_i}$ is the 3×3 diagonal matrix of the mass of link i:

$$\mathbf{M_i} = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \tag{4.2}$$

and $\bar{\mathbf{I}}_\mathbf{i}$ is the 3×3 inertia tensor of link i about the origin of the ith coordinate frame. The 3×3 matrix $\tilde{\mathbf{h}}_\mathbf{i}$ is defined by:

$$\tilde{\mathbf{h}}_\mathbf{i} = m_i \tilde{\mathbf{s}}_\mathbf{i}, \tag{4.3}$$

where $\mathbf{s_i}$ is the position vector of the center of gravity of link i from the origin of coordinate frame i. Because the coordinate frames are fixed to their respective links and the links are rigid, the inertia matrix $\mathbf{I}_\mathrm{i}$ is constant for all link orientations.

The transformation of a spatial inertia matrix to an adjacent coordinate frame can be done as follows [2, 20]:

$$ {}^{\mathrm{i}}\mathbf{I}_{\mathrm{i+1}} = {}^{\mathrm{i+1}}\mathbf{X}_{\mathrm{i}}^{\mathrm{T}}\; {}^{\mathrm{i+1}}\mathbf{I}_{\mathrm{i+1}}\; {}^{\mathrm{i+1}}\mathbf{X}_{\mathrm{i}} \tag{4.4} $$

4.3 The motion of links in a manipulator

The motion of one link relative to another is constrained by the joint that connects them. As such, the relative velocity and acceleration of two links is only a function of the joint space and joint velocity and acceleration of this joint. Since the motion of the base is known, the velocity of the first link can be determined if the joint velocity of the first joint is known, and similarly the acceleration. If the first link's motion is known, and the relative motion of the second link is a function of the joint velocity and acceleration of the second joint, the motion of the second link can be determined. By continuing in this way to the last link, the motions of all the links can be determined.

Let us first consider the relative motions. The spatial velocity of link i relative to link $(i-1)$, expressed in the i-th coordinate frame is simply:

$$ \phi_i \dot{q}_i^*, \tag{4.5} $$

where $\dot{q}_i^*$ is the joint velocities of joint i and ϕ_i is the motion space of joint i. The spatial velocity of link $(i-1)$, which is expressed in link frame $(i-1)$, can be transformed to frame i to find the spatial velocity of frame i due to the velocity of link $(i-1)$:

$$ {}^{\mathrm{i}}\mathbf{X}_{\mathrm{i-1}}\mathbf{v}_{\mathrm{i-1}}, \tag{4.6} $$

where $\mathbf{v}_{\mathrm{i-1}}$ is the spatial velocity of link $(i-1)$ (the spatial velocity of link frame $(i-1)$), and ${}^{\mathrm{i}}\mathbf{X}_{\mathrm{i-1}}$ is the spatial transformation from frame $(i-1)$ to frame i. Given the spatial velocity of link $(i-1)$ and the joint velocity of joint i, the spatial velocity of link i, expressed in frame i is:

$$ \mathbf{v}_{\mathrm{i}} = {}^{\mathrm{i}}\mathbf{X}_{\mathrm{i-1}}\mathbf{v}_{\mathrm{i-1}} + \phi_i \dot{q}_i^*. \tag{4.7} $$

The spatial transformation already accounts for the linear velocity of the i-th frame due to the angular velocity of frame (and link) i. To show this, equation 4.7 can be expanded and rewritten:

$$ \begin{aligned} \mathrm{v}_{\mathrm{i}} &= \begin{bmatrix} {}^{i}\mathrm{A}_{i-1} & 0 \\ {}^{i}\mathrm{A}_{i-1}(\tilde{\mathrm{b}}_i)^T & {}^{\mathrm{i}}\mathrm{A}_{\mathrm{i-1}} \end{bmatrix} \begin{bmatrix} \omega_{i-1} \\ \mathrm{u}_{i-1} \end{bmatrix} + \phi_i \dot{q}_i^* && (4.8) \\ &= \begin{bmatrix} {}^{i}\mathrm{A}_{i-1}\omega_{i-1} \\ {}^{i}\mathrm{A}_{i-1}(\omega_{i-1} \times \mathrm{b}_i + \mathrm{u}_{i-1}) \end{bmatrix} + \phi_i \dot{q}_i^* && (4.9) \end{aligned} $$

The spatial acceleration of link i can also be written as the sum of the relative acceleration due to joint i and the spatial acceleration of link $(i-1)$, but one has to add another term to account for the bias accelerations since the i-th coordinate frame is not an inertial one. Let ζ_i be the spatial bias acceleration vector. A recursive relationship to determine the spatial acceleration of link i from the spatial acceleration of link $(i-1)$ and the joint acceleration of joint i is then simply:

$$a_i = {}^iX_{i-1}a_{i-1} + \phi_i\ddot{q}_i^* + \zeta_i \tag{4.10}$$

Consider the bias acceleration vector. As is well known, for non-inertial systems, the bias linear acceleration consists of the centripetal and Coriolis accelerations. The centripetal acceleration (linear) of the origin of the ith link frame relative to the origin of frame $(i-1)$ is given by:

$$\begin{aligned} \mathrm{a}_{centripetal} &= {}^i\mathbf{A}_{i-1}\left[\omega_{i-1} \times (\omega_{i-1} \times \mathbf{b}_i)\right], & (4.11)\\ &= {}^i\mathbf{A}_{i-1}\left[\tilde{\omega}_{i-1}(\tilde{\omega}_{i-1}\mathbf{b}_i)\right], & (4.12) \end{aligned}$$

where the tensor dual of angular velocity is used to accomplished the cross product. The Coriolis acceleration (linear) of frame i is:

$$\begin{aligned} \mathrm{a}_{Coriolis} &= 2\,{}^i\mathbf{A}_{i-1}\left[\omega_{i-1} \times \mathbf{u}_i^r\right], & (4.13)\\ &= 2\,{}^i\mathbf{A}_{i-1}\left[\tilde{\omega}_{i-1}\mathbf{u}_i^r\right], & (4.14) \end{aligned}$$

where $\mathbf{u}_i^r$ is the linear velocity of frame i relative to frame $(i-1)$, which is only non-zero if joint i has a prismatic degree of freedom. The relative linear velocity is given by the last three components of $\phi_i\dot{q}_i^*$ and it can be written as:

$$\mathbf{u}_i^r = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \phi_i\dot{q}_i^*. \tag{4.15}$$

The bias spatial acceleration can also have rotational components. If the angular velocity of link $(i-1)$ and link i is not parallel, there exists a bias angular acceleration:

$$\begin{aligned} \dot{\omega}_i^{bias} &= {}^i\mathbf{A}_{i-1}\omega_{i-1} \times \omega_i^r, & (4.16)\\ &= {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1}\omega_i^r. & (4.17) \end{aligned}$$

ω_i^r is the angular velocity of link i relative to link $(i-1)$ and is given by the first three components of $\phi_i\dot{q}_i^*$:

$$\omega_i^r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \phi_i\dot{q}_i^*. \tag{4.18}$$

By using the above equation that describes the components of the bias acceleration vector, the spatial acceleration of link i becomes:

$$\mathrm{a_i} = {}^{\mathrm{i}}\mathrm{X_{i-1}}\mathrm{a_{i-1}}\phi_i\ddot{q}_i^* + \begin{bmatrix} {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} & 0 \\ 0 & 2\,{}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} \end{bmatrix}\phi_i\dot{q}_i^* + \begin{bmatrix} 0_{3\times1} \\ {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1}\tilde{\omega}_{i-1}\mathbf{b}_i \end{bmatrix}. \tag{4.19}$$

If the SDOF kinematic notation is used, equation 4.19 becomes:

$$\mathrm{a_i} = {}^{\mathrm{i}}\mathrm{X_{i-1}}\mathrm{a_{i-1}}\phi_i\ddot{q}_i^* + \begin{bmatrix} {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} & 0 \\ 0 & 2\,{}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} \end{bmatrix}\phi_i\dot{q}_i^* + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1}\tilde{\omega}_{i-1} \end{bmatrix}\gamma_{i-1}. \tag{4.20}$$

4.4 The forces acting on links in a manipulator

In order to determine what forces act on each link in the system, each link can be considered as a free body, and Newton and Euler's equations of motion applied to these free bodies. From these equations the resultant spatial force on each link can be found. The only forces that can act on the links (excluding the last link) are the forces that neighbouring links exert on them[4]. In a serial chain, the inner links thus has two unknown forces acting on it of which the resultant is known. There exists infinite solutions for these two forces that will have the required resultant. Applying the equations of motion to an arbitrary link in the chain can thus be interesting, but not useful by worldly standards. The obvious solution is to first apply the equations of motion to the last link in the serial chain (or subchain)[5]. The forces acting on the last link can only be the force exerted on it by the previous link and the force exerted on it by the environment[6]. There is only one solution for the force exerted on the last link by the previous link that will satisfy the resultant force requirement. This unique solution can be found, and the image of this force on the motion space of the joint is the joint forces. Since the force that the i-th link exerts on the $(i-1)$-th link is exactly opposite to the force that the $(i-1)$-th link exert on the i-th link, one of the forces acting on the previous link is known. The other force can be calculated using the equations of motion. In this way the procedure can be iterated towards

[4] Gravity also exerts a force on each link. Calculating the force exerted on each link to take the influence of gravity into account is not very efficient (although it is optimally effective). A more efficient way is to give the base of the system an acceleration equal but opposite to gravitational acceleration.

[5] Remember that multiple chain mechanisms can be divided into parts that are serial chains.

[6] If the chain was part of a bigger chain, this environment would be the connection with the rest of the mechanism. The force in this connection is not known, but one can choose an appropriate value for this force, or in the case of a simple closed chain mechanism, this force can be calculated as in Chapter 6 since the required acceleration of the reference member is given.

the base of the chain until all the joint forces have been calculated.

The equations of motion are combined into a single equation with the use of spatial notation. The equation of motion for the i-th link at the link frame origin and expressed in the link frame is [2]:

$$\mathbf{F_i} = \mathbf{I_i a_i} + \begin{bmatrix} \tilde{\omega}_i \bar{\mathbf{I}}_i \omega_i \\ m_i \tilde{\omega}_i (\tilde{\omega}_i \mathbf{s}_i) \end{bmatrix}. \tag{4.21}$$

$\mathbf{s}_i$ is the position of the center of mass of link i relative to the origin of link frame i, with components expressed in frame i. Now we will express the resultant force on link i in terms of the two forces acting on the link. Force $\mathbf{f_{i+1}}$ is exerted on the near end of link $(i+1)$, and the reaction acts on the far end of link i. The sum of this force, transformed to the near end of link i, and the force acting on joint i (in both the constrained and unconstrained directions) is the resultant force acting on link i. Unfortunately, our definition of the 6×1 force vector does not satisfy the general spatial transformation given in equation 2.5. The transformation used below should be easy to verify. The resultant force is thus:

$$\mathbf{F_i} = \mathbf{f_i} + \begin{bmatrix} {}^i\mathbf{A}_{i+1} & \tilde{\mathbf{b}}_{i+1}\, {}^i\mathbf{A}_{i+1} \\ \mathbf{0} & {}^i\mathbf{A}_{i+1} \end{bmatrix} (-\mathbf{f_{i+1}}). \tag{4.22}$$

This is the resultant force used in the equation of motion (equation 4.21). Equation 4.22 can be rearranged and the resultant force calculated with equation 4.21 substituted into it to obtain the force on the near end of link i:

$$\mathbf{f_i} = \mathbf{F_i} + \begin{bmatrix} {}^i\mathbf{A}_{i+1} & \tilde{\mathbf{b}}_{i+1}\, {}^i\mathbf{A}_{i+1} \\ \mathbf{0} & {}^i\mathbf{A}_{i+1} \end{bmatrix} \mathbf{f_{i+1}}. \tag{4.23}$$

As mentioned before, the components of $\mathbf{f_i}$ that lies in the joint space of joint i is the joint force for joint i. The other components is the constraint force in joint i. Thus,

$$\tau_i^* = \phi_i^T \mathbf{f_i} \tag{4.24}$$

If friction is present in the joint, the friction force has to be added to the above equation:

$$\tau_i^* = \phi_i^T \mathbf{f_i} + \mathcal{F}(\dot{q}_i^*), \tag{4.25}$$

where $\mathcal{F}$ depends on the type of friction that is modeled and can be a non-linear function. For viscous friction, it would simply be:

$$\mathcal{F} = b_i \dot{q}_i^*, \tag{4.26}$$

where b_i is a vector of (positive) friction components for the free axis of the joint.

The algorithm for determining the joint forces required to achieve a prescribed manipulator motion is summarized below:

4.4.1 Recursive algorithm for solving the inverse dynamics of manipulators with multiple degree-of-freedom joints

1. Given: v_0 and a_0
   Initialize: $\Gamma = [\,]$

2. Forward equations: $i = 1, 2, \ldots, N$

$$\mathrm{v_i} = {}^{\mathrm{i}}\mathrm{X_{i-1}} \mathrm{v_{i-1}} + \phi_i \dot{q}_i^* \tag{4.27}$$

$$\mathrm{a_i} = {}^{\mathrm{i}}\mathrm{X_{i-1}} \mathrm{a_{i-1}} \phi_i \ddot{q}_i^* + \begin{bmatrix} {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} & 0 \\ 0 & 2\,{}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1} \end{bmatrix} \phi_i \dot{q}_i^* + \begin{bmatrix} 0_{3\times 1} \\ {}^i\mathbf{A}_{i-1}\tilde{\omega}_{i-1}\tilde{\omega}_{i-1}\mathrm{b}_i \end{bmatrix} \tag{4.28}$$

3. Backward equations: $i = N, N-1, \ldots, 1$

$$\mathrm{F_i} = \mathrm{I_i a_i} + \begin{bmatrix} \tilde{\omega}_i \bar{\mathrm{I}}_i \omega_i \\ m_i \tilde{\omega}_i (\tilde{\omega}_i \mathrm{s}_i) \end{bmatrix} \tag{4.29}$$

$$\mathrm{f_i} = \mathrm{F_i} + \begin{bmatrix} {}^i\mathbf{A}_{i+1} & \tilde{\mathrm{b}}_{i+1}\,{}^i\mathbf{A}_{i+1} \\ 0 & {}^i\mathbf{A}_{i+1} \end{bmatrix} \mathrm{f_{i+1}} \tag{4.30}$$

$$\tau_i^* = \phi_i^T \mathrm{f_i} + \mathcal{F}(\dot{q}_i^*), \tag{4.31}$$

$$\Gamma = [\tau_i^* \;\; \Gamma] \tag{4.32}$$

The usual base conditions are: $\mathrm{v}_0 = \mathbf{0}$ and $\mathrm{a}_0 = \begin{bmatrix} 0_{3\times 1} \\ -\mathrm{g} \end{bmatrix}$, where g is the gravitational acceleration vector.

4.5 Worked example

Here we consider the application of the inverse dynamics algorithm to the "Stanford manipulator". We now have to give the manipulator dimensions and inertial properties. The dimensions are chosen as in Figure 4.1.

The manipulator's structure and kinematic properties is described by the SDOF joint parameters and the joint spaces. The joint parameters can be found as explained in Chapter 2. We have already described the joint spaces. The joint parameters and joint spaces are summarized in Table 4.1.

The inertial properties of the links will now be assigned. We have to specify the mass of each link, the position of the center of gravity of each link relative to the

Link i	Joint parameters γ_i	Joint space ϕ_i
1	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0,5 \\ -0,15 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0,8 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
3	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0,2 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
4	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0,05 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Table 4.1: The SDOF joint parameters and joint spaces of the Stanford manipulator.

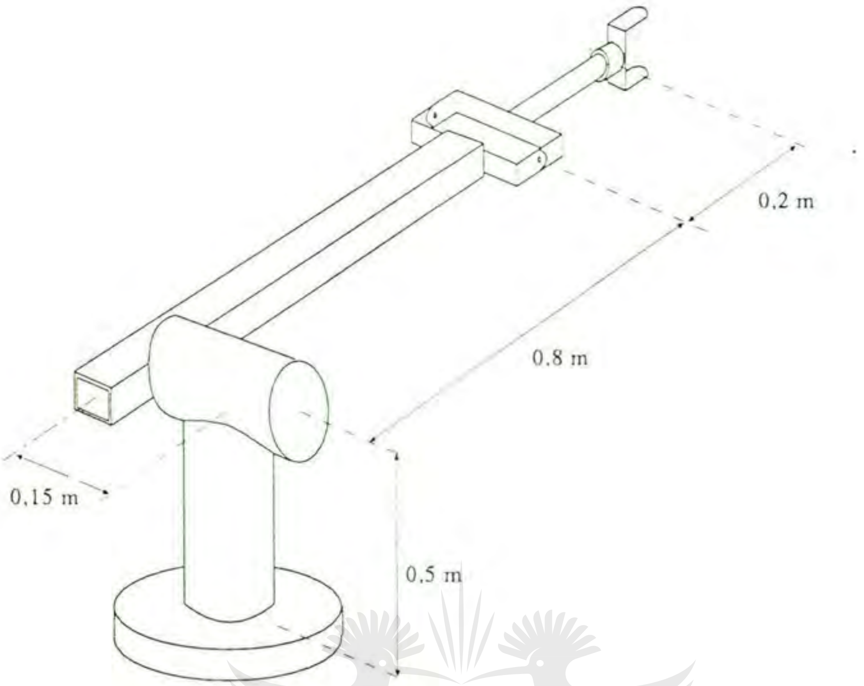


Figure 4.1: The dimensions of our Stanford manipulator.

link coordinate frame, and the inertia tensor of each link around its center of gravity. From this we can calculate the spatial inertia of each link around their respective link frame origins. The positions and orientations of the link frames are still as in Figure 2.5. Let the first link have a mass of $2\,kg$, a center of mass $0,3\,m$ above the base and $0,07\,m$ to the left of the base:

$$m_1 = 2\,kg \tag{4.33}$$

$$\mathrm{s}_1 = \left[\begin{array}{c} 0 \\ 0,3 \\ -0,07 \end{array}\right] m \tag{4.34}$$

The inertia tensor is:

$$\underline{\mathbf{I}}_i = \left[\begin{array}{ccc} 0,1 & 0 & 0 \\ 0 & 0,02 & 0 \\ 0 & 0 & 0,1 \end{array}\right] kg\,m^2 \tag{4.35}$$

The other links' dynamic properties are summarized in Table 4.2. For all the links, the inertia tensor around the link frame origin can be calculated by using the parallel axis theorem. For link 1 this is done as follows:

$$\bar{\mathbf{I}}_1 = \underline{\mathbf{I}}_1 - m_1\tilde{\mathrm{s}}_1\tilde{\mathrm{s}}_1 \tag{4.36}$$

Link	Mass	Position of center of mass	Inertia Tensor
2	2	$\begin{bmatrix} 0,3 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,002 & 0 & 0 \\ 0 & 0,25 & 0 \\ 0 & 0 & 0,25 \end{bmatrix}$
3	0,8	$\begin{bmatrix} 0,12 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,002 & 0 & 0 \\ 0 & 0,008 & 0 \\ 0 & 0 & 0,008 \end{bmatrix}$
4	0,5	$\begin{bmatrix} 0,05 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,001 & 0 & 0 \\ 0 & 0,004 & 0 \\ 0 & 0 & 0,004 \end{bmatrix}$

Table 4.2: The inertial properties of links 2 to 4.

$$\begin{aligned} &= \begin{bmatrix} 0,1 & 0 & 0 \\ 0 & 0,02 & 0 \\ 0 & 0 & 0,1 \end{bmatrix} - 2 \begin{bmatrix} 0 & 0,07 & 0,3 \\ -0,07 & 0 & 0 \\ -0,3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0,07 & 0,3 \\ -0,07 & 0 & 0 \\ -0,3 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0,2898 & 0 & 0 \\ 0 & 0,0298 & 0,042 \\ 0 & 0,042 & 0,28 \end{bmatrix} \end{aligned} \tag{4.37}$$

Now the spatial inertia of link 1 can be calculated:

$$\mathbf{I}_1 = \begin{bmatrix} \bar{\mathbf{I}}_i & m_1\tilde{\mathbf{s}}_1 \\ m_1\tilde{\mathbf{s}}_1 & m_1\mathbf{1}_{3\times 3} \end{bmatrix}, \tag{4.38}$$

$$= \begin{bmatrix} 0,2898 & 0 & 0 & 0 & 0,14 & 0,6 \\ 0 & 0,0298 & 0,042 & -0,14 & 0 & 0 \\ 0 & 0,042 & 0,28 & -0,6 & 0 & 0 \\ 0 & -0,14 & -0,6 & 2 & 0 & 0 \\ 0,14 & 0 & 0 & 0 & 2 & 0 \\ 0,6 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}. \tag{4.39}$$

The spatial inertias of the other links are as follows:

$$\mathbf{I}_2 = \begin{bmatrix} 0{,}002 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}43 & 0 & 0 & 0 & -0{,}6 \\ 0 & 0 & 0{,}43 & 0 & 0{,}6 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0{,}6 & 0 & 2 & 0 \\ 0 & -0{,}6 & 0 & 0 & 0 & 2 \end{bmatrix}, \tag{4.40}$$

$$\mathbf{I}_3 = \begin{bmatrix} 0{,}002 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}01952 & 0 & 0 & 0 & -0{,}096 \\ 0 & 0 & 0.01952 & 0 & 0{,}096 & 0 \\ 0 & 0 & 0 & 0{,}8 & 0 & 0 \\ 0 & 0 & 0{,}096 & 0 & 0{,}8 & 0 \\ 0 & -0{,}096 & 0 & 0 & 0 & 0{,}8 \end{bmatrix}, \tag{4.41}$$

$$\mathbf{I}_4 = \begin{bmatrix} 0{,}001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}00525 & 0 & 0 & 0 & -0{,}025 \\ 0 & 0 & 0{,}00525 & 0 & 0{,}025 & 0 \\ 0 & 0 & 0 & 0{,}5 & 0 & 0 \\ 0 & 0 & 0{,}025 & 0 & 0{,}5 & 0 \\ 0 & -0{,}025 & 0 & 0 & 0 & 0{,}5 \end{bmatrix}. \tag{4.42}$$

All the inertial properties of the manipulator are now known. Now we can consider the motion that the manipulator has to undergo. To make the example interesting, we will let joint one and the prismatic joint both have a constant velocity. This will demonstrate the centripetal and Coriolis effects, as well as the influence of gravity. Joint three and four will also be given rotational joint velocities, leading to interesting effects that will be discussed later. The vector of joint velocities[7] is assigned as:

$$\dot{\mathrm{q}} = \begin{bmatrix} 1{,}5 \\ 0 \\ 0{,}4 \\ 1 \\ 3 \end{bmatrix}. \tag{4.43}$$

To illustrate only inertial and gravitational effects, all the joint accelerations are set to zero:

$$\ddot{\mathrm{q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.44}$$

[7] For rotational degrees-of-freedom the joint velocities are expressed in *radian per second*, and for the prismatic components in *meters per second*.

The inverse dynamics algorithm of Section 4.4.1 can now be applied. For the current configuration of the manipulator, all the rotation transformations are identity matrices:

$$ {}^{i}\mathbf{A}_{i-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{for } i = 1, \ldots, 4. \tag{4.45}$$

The spatial transformations are:

$$ {}^{1}\mathbf{X}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.46}$$

$$ {}^{2}\mathbf{X}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -0,15 & -0,5 & 1 & 0 & 0 \\ 0,15 & 0 & 0 & 0 & 1 & 0 \\ 0,5 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.47}$$

$$ {}^{3}\mathbf{X}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0,8 & 0 & 1 & 0 \\ 0 & -0,8 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.48}$$

$$ {}^{4}\mathbf{X}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0,2 & 0 & 1 & 0 \\ 0 & -0,2 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.49}$$

Equation 4.27 is used to calculate the spatial velocities of the link, while equation 4.28 is used to calculate the spatial accelerations of the links. The results are as follow:

$$ \mathrm{v}_1 = \begin{bmatrix} 0 \\ 1,5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathrm{a}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 9.81 \\ 0 \end{bmatrix}. \tag{4.50}$$

$$\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1{,}5 \\ 0 \\ 0{,}175 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 9.81 \\ -0{,}8625 \end{bmatrix}, \tag{4.51}$$

$$\mathbf{v}_3 = \begin{bmatrix} 0 \\ 1.5 \\ 1 \\ 0{,}175 \\ 0 \\ -1{,}2 \end{bmatrix}, \quad \mathbf{a}_3 = \begin{bmatrix} 1.5 \\ 0 \\ 0 \\ -1{,}8 \\ 9{,}81 \\ -0{,}8625 \end{bmatrix}, \tag{4.52}$$

$$\mathbf{v}_4 = \begin{bmatrix} 3 \\ 1{,}5 \\ 1 \\ 0{,}175 \\ 0{,}2 \\ -1{,}5 \end{bmatrix}, \quad \mathbf{a}_4 = \begin{bmatrix} 1.5 \\ 3 \\ -4{,}5 \\ -2{,}45 \\ 9{,}81 \\ -0{,}8625 \end{bmatrix}. \tag{4.53}$$

Note that each link has a linear acceleration component in the y-direction. It is due to the acceleration we gave the base to take gravity into account. The last component of $\mathbf{a}_2$ is the centripetal and Coriolis acceleration of the coordinate frame of link 2 (the Coriolis acceleration is much larger and in the opposite direction to the centripetal acceleration). It has a centripetal acceleration since coordinate frame 2 does not lie on the axis of joint 1. The spatial acceleration of link 3 is very interesting. None of the joints have rotational acceleration components, yet link 3 has an angular acceleration of 1,5 around the x-axis. This is because the rotational axes of joint 2 and 3 are not parallel. As seen in equation 4.17, this results in an angular acceleration: the direction of the rotation axis of joint 3 changes as joint 2 rotates – hence an angular acceleration. Link three also has this angular acceleration, but in addition to that it also has angular accelerations in the y and z-directions because the resultant angular velocity of the end of link 3 has components in z (due to joint 2 and 3) and y (due to joint 1), while joint 4 rotates about its x axis. Again these rotations are not parallel, leading to the additional angular acceleration.

The acceleration of all the links, expressed in their respective link coordinate frames (which, by coincidence, are all parallel) are now known. Equation 4.29 can now be used to calculate the resultant spatial forces on the links, also expressed in the respective link coordinate frames. These forces are:

$$\mathbf{F}_4 = \begin{bmatrix} 0,0015 \\ 0.0245625 \\ 0.24075 \\ -1.30625 \\ 4.905 \\ -0.43125 \end{bmatrix}, \tag{4.54}$$

$$\mathbf{F}_3 = \begin{bmatrix} 0,003 \\ 0,0828 \\ 0.94176 \\ -1,752 \\ 7,848 \\ -0,69 \end{bmatrix}, \tag{4.55}$$

$$\mathbf{F}_2 = \begin{bmatrix} 0 \\ 0,5175 \\ 5,886 \\ -1,35 \\ 19,62 \\ -1,725 \end{bmatrix}, \tag{4.56}$$

$$\mathbf{F}_1 = \begin{bmatrix} 1,4679 \\ 0 \\ 0 \\ 0 \\ 19,62 \\ 0,315 \end{bmatrix}. \tag{4.57}$$

No force is applied to the tip of the manipulator. The spatial force applied by joint 4 on link 4 must thus equal the resultant force calculated above. By using a backward recursion, the forces on the near end of the links (closest to the base) can be calculated (equation 4.30). The components of these forces that lies in the motion space of the joints are the applied joint forces (equation 4.31). The calculated forces and joint forces are given below:

$$\mathbf{f}_4 = \begin{bmatrix} 0.0015 \\ 0,0245625 \\ 0,24075 \\ -1,30625 \\ 4,905 \\ -0.43125 \end{bmatrix}, \quad \tau_4^* = \begin{bmatrix} 0,0015 \end{bmatrix}, \tag{4.58}$$

$$\mathrm{f}_3 = \begin{bmatrix} 0,0045 \\ 0,1936125 \\ 2,16351 \\ -3,05825 \\ 12,753 \\ -1,12125 \end{bmatrix}, \quad \tau_3^* = \begin{bmatrix} 2,16351 \end{bmatrix}, \tag{4.59}$$

$$\mathrm{f}_2 = \begin{bmatrix} 0,0045 \\ 1,6081125 \\ 18,25191 \\ -4,40825 \\ 32,373 \\ -2,84625 \end{bmatrix}, \quad \tau_2^* = \begin{bmatrix} 18,25191 \\ -4,40825 \end{bmatrix}, \tag{4.60}$$

$$\mathrm{f}_1 = \begin{bmatrix} 4,905225 \\ 2,26935 \\ 20,456035 \\ -4,40825 \\ 51,993 \\ -2,53125 \end{bmatrix}, \quad \tau_1^* = \begin{bmatrix} 2,26935 \end{bmatrix}. \tag{4.61}$$

τ_1^* is largely due to the Coriolis acceleration of link two. The first component of τ_2^* is due to gravitational forces of links 2 to 4. The second component is purely due to the centripetal acceleration of links 2 to four, which is a result of the joint velocity of joint 1. Both τ_3^* and τ_4^* is due solely to bias angular accelerations. The vector of required applied joint forces are thus:

$$\tau = \begin{bmatrix} 2,26935 \\ 18,25191 \\ -4,40825 \\ 2,16351 \\ 0,0015 \end{bmatrix}. \tag{4.62}$$

Now one can appreciate the complexity that arises when rigid bodies are connected to one another. The motion of any one of them can have an influence on all the other bodies in the system. The forces that result from these interactions (especially the bias forces) are often hard to see intuitively. Often these bias forces are large in magnitude – even for relatively slow motions – and cannot be ignored.

4.6 Solution example

We would now like to apply our imaginary Stanford manipulator to something one can imagine to be useful. We want the manipulator to do the following:

On ground level (the level of the base coordinate frame), at $0,5\,m$ from the base in the z-direction, there is a tap that the manipulator has to close by half a turn. First, the manipulator has to position itself so that the gripper fits over the tap handle. The axis around which the gripper can turn has to be vertical. Because of the current position of the tap handle, the gripper also has to turn 90^o clockwise before it can engage the tap handle. Next, the manipulator must turn the handle (and thus its gripper) 180^o clockwise. It has been determined that to overcome the resistance of the tap handle to being turned requires $5\,Nm$. The gripper thus has to exert a moment of $5\,Nm$ (clockwise) on the tap handle to turn it. It is assumed that the rotational moments of inertia of the tap handle and connected parts is small compared to the inherent friction in the tap mechanicals.

We know the manipulator position at the start and end of the task, but we need to know what forces must be applied to the joints to accomplish the task. The manipulator motion has not been completely specified yet, only the manipulator positions at three instants have been given. Consider the first motion from the initial position to the position on the tap handle. We would like the manipulator to have no velocity at the start of the motion to the work position, and at the work position. An interpolation function that can satisfy these requirements is a simple third order polynomial. For all the joint positions we will fit a third order polynomial between the initial and work position states. Let the manipulator accomplish this motion in one second. We will do the same for the motion of the manipulator while turning the tap, but allow only half a second for turning the tap through 180^o. A graph of the joint positions is given in Figure 4.2. Once these joint position functions has been calculated, the joint velocity and acceleration functions can also be calculated. The results are shown in Figures 4.3 and 4.4. The motion of the manipulator has then been sufficiently (actually completely) specified. The force that the end effector (gripper) has to exert is also known as function of time (Figure 4.5). The inverse dynamics algorithm can now be applied to determine the joint forces. Friction in the manipulator will again be ignored.

The detail concerning the application of the inverse dynamics algorithm will not be given here (see previous section), but only some important points will be highlighted. The first is that the position of the center of mass of link two is not constant relative to the link frame, but it a function of the displacement of the prismatic joint. It is a simple linear function of the displacement of the prismatic joint and is easily calculated. Let the link be a simple beam of length l, with the position of its center of mass relative to the near end given by **cg**. The relationship between the position

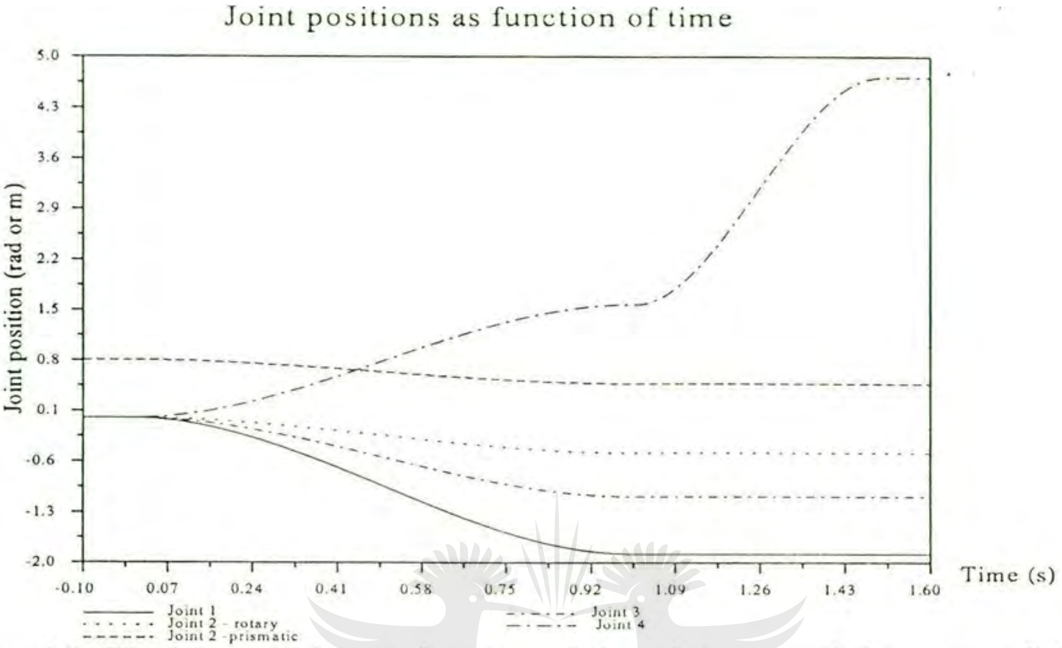


Figure 4.2: The joint positions as functions of time that are needed to accomplish the task (closing a valve by half a turn).

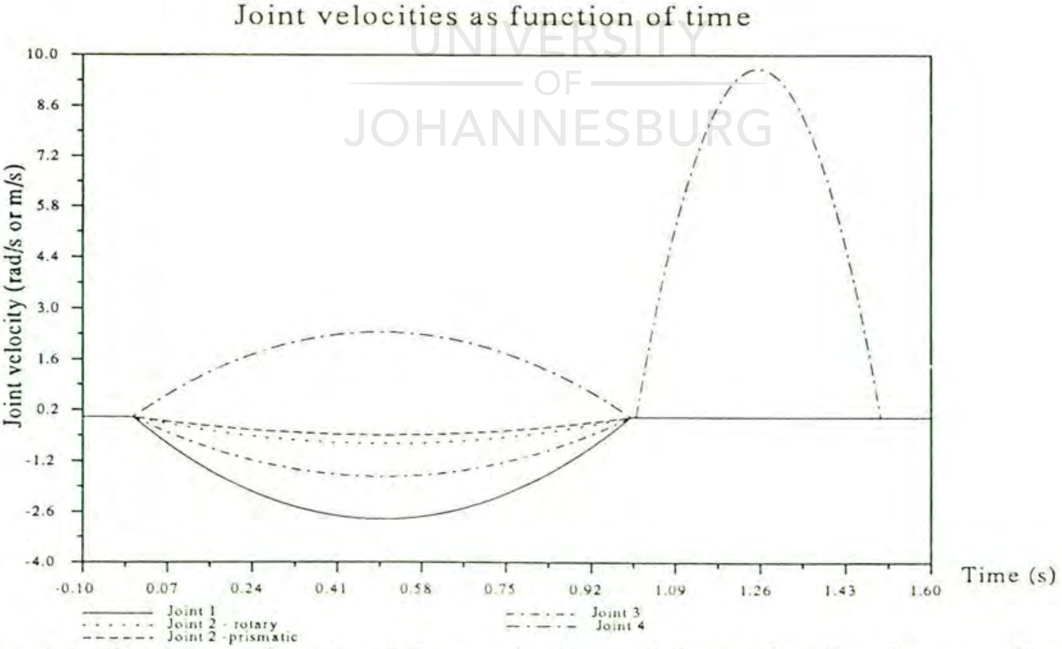


Figure 4.3: The joint velocities of the manipulator whilst performing the prescribed task.

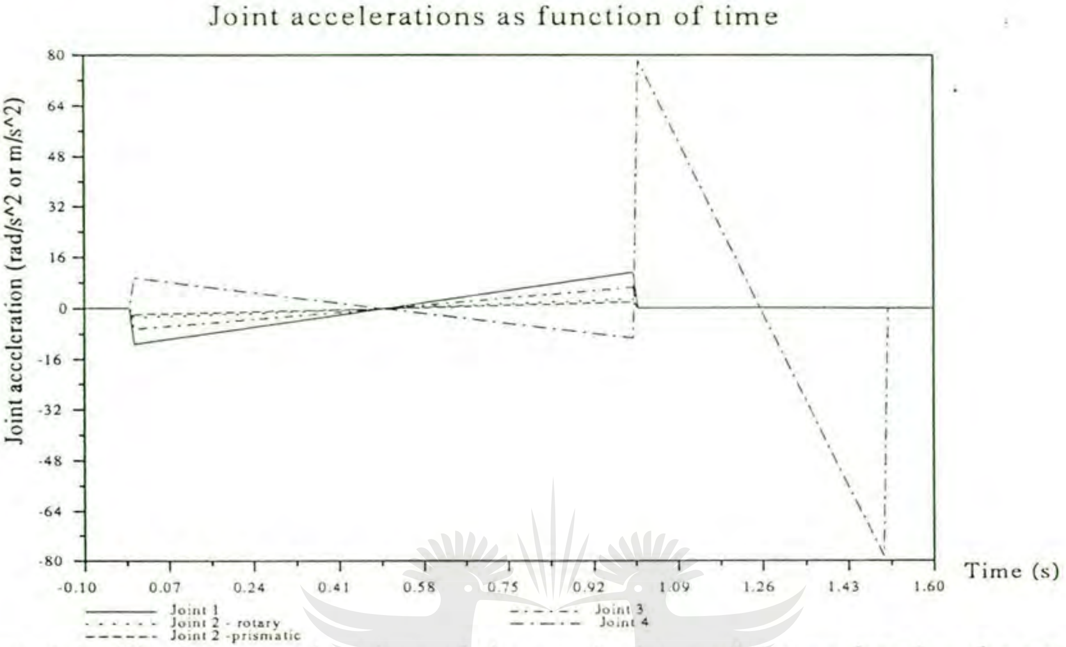


Figure 4.4: The joint accelerations of the manipulator whilst performing the prescribed task.

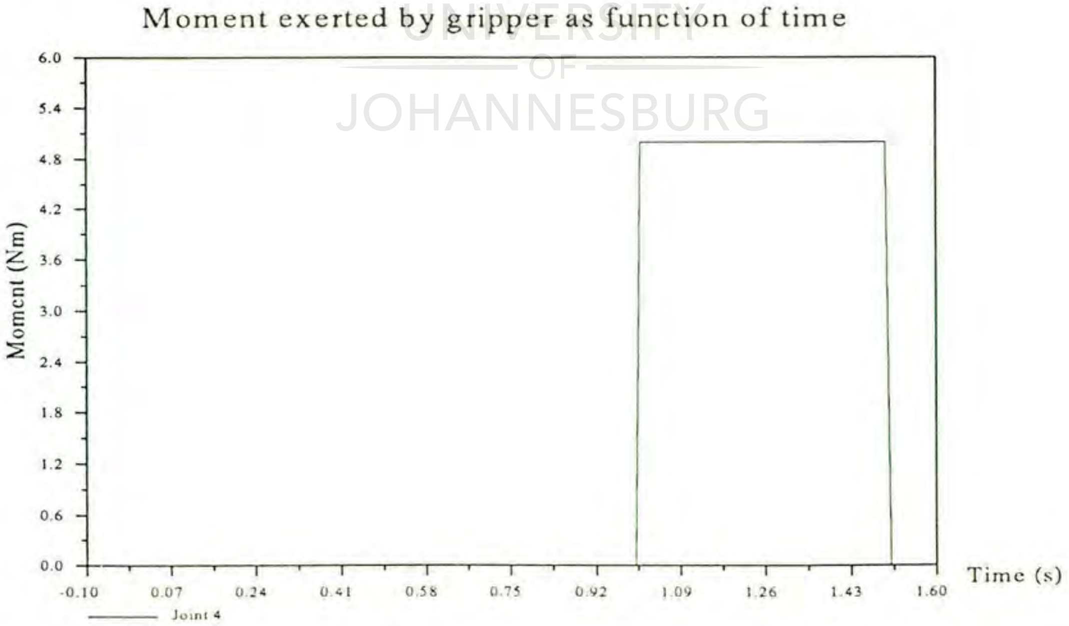


Figure 4.5: The force exerted by the gripper whilst performing the prescribed task.

of the center of mass and the prismatic displacement is then simply:

$$\mathrm{s}_2 = ([0\ 1]q_2^* - l)\begin{bmatrix}1\\0\\0\end{bmatrix} + \mathrm{cg}. \tag{4.63}$$

For each time step, the new position of the center of mass of link 2 relative to link frame two has to be calculated, and from this the spatial inertia of link two around the link frame has to be recalculated. An implementation of the solution to the motion problem described here is the Scilab [22] executable *stanford_moving.exe*, which is listed in the appendices. The joint forces needed to accomplish the task is calculated by the algorithm and the results are given in Figure 4.6. Notice the change in joint forces the moment the gripper starts applying a force to the tap. Another insight one can gather from the graph is the influence bias forces and forces due to the interaction of the links. The joint forces look considerably different to the joint accelerations of Figure 4.4.

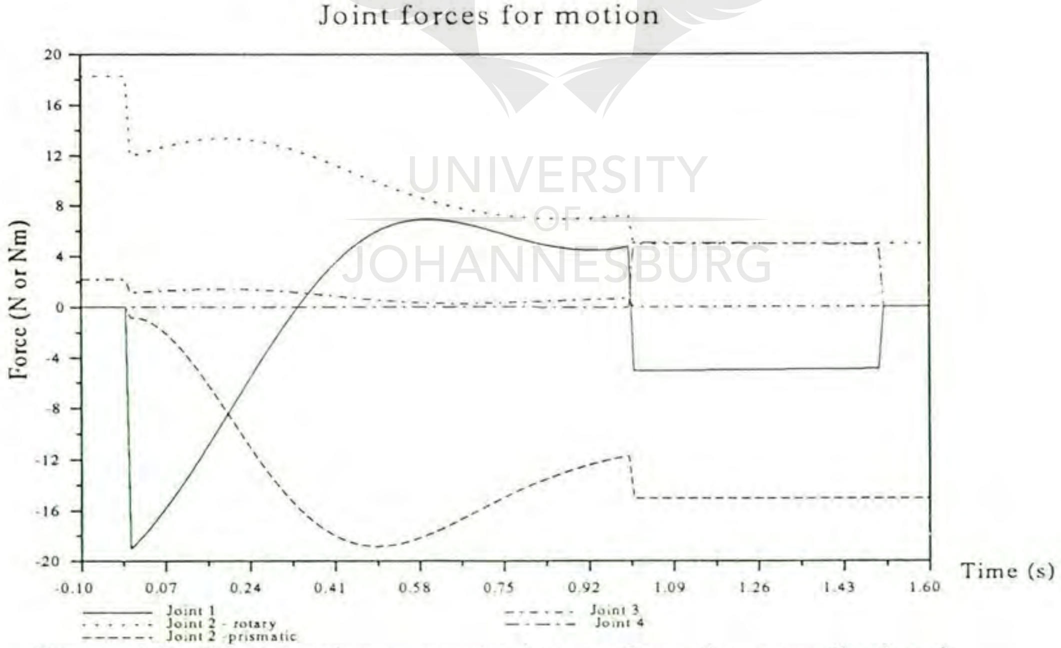


Figure 4.6: The joint forces required to perform the prescribed task.

Chapter 5

Direct dynamics of serial chains

5.1 Introduction

The direct dynamics problem, also called the forward dynamics problem,[1] is concerned with determining the joint accelerations in a robotic system when the applied joint forces are given. The solutions of the direct dynamic problem is most frequently applied in the simulation of robotic systems for the testing of control systems and the training of adaptive controllers, and for providing state feedback in tele-operated systems. In this chapter we will only consider single serial chain manipulators. It is the simplest sub class of manipulators that the direct dynamic problem can be solved for, but the importance of these solutions must not be underestimated for several reasons:

- Many real systems can be modeled as single serial chains.
- The results for serial chains can be used extensively in the analysis of more complex systems (see Chapter 6).

Single closed chains are divided into two classes: open-chains and closed-chains. When only one end of a serial chain is connected to, or in contact with a base, this manipulator is in a serial open-chain configuration. If the other end is also connected or in contact with a base, the manipulator is in a closed-chain configuration. In this chapter the spatial notation will be used in the dynamic analyses. Much work has been done by other researchers, and the analysis methods that follow are mostly a summary of the work by Lilly [2], with some comments and explanations.

[1] This is different to the kinematic problem where the forward problem of kinematics corresponds closer to the inverse problem of dynamics.

5.2 Equations of direct dynamics

Various methods can be used to derive the equations of motion for serial manipulators. The most commonly used in robotics are the Lagrange-Euler and the Newton-Euler methods. With these methods dynamic equations can be formulated in terms of either the *operational space*[2] or the *joint space*[3].

The operational space formulation is useful when only the motion of the end effector needs to be studied. When the operational space formulation is used, the dynamic behavior of the manipulator is described in terms of its end effector. The resulting equations can be manipulated to have the following form for an open chain [2, chapter 4]:

$$\mathrm{F} = \Lambda(\mathrm{x})\ddot{\mathrm{x}} + \mu(\mathrm{x}, \dot{\mathrm{x}}) + \mathrm{p}(\mathrm{x}) \tag{5.1}$$

where

$$\begin{aligned}
\mathrm{F} &= 6 \times 1 \text{ operational force vector,} \\
\Lambda(\mathrm{x}) &= 6 \times 6 \text{ operational space inertia matrix,} \\
\mathrm{x} &= 6 \times 1 \text{ vector of end effector coordinates,} \\
\dot{\mathrm{x}}, \ddot{\mathrm{x}} &= 6 \times 1 \text{ spatial vectors of end effector rates and accelerations,} \\
\mu(\mathrm{x}, \dot{\mathrm{x}}) &= 6 \times 1 \text{ vector of centripetal and Coriolis forces, and} \\
\mathrm{p}(\mathrm{x}) &= 6 \times 1 \text{ vector of gravity forces.}
\end{aligned}$$

Λ is an inertial quantity which combines the properties of the entire chain and projects them to the tip or end effector.

In a similar way, the joint space dynamic equations of motion for a single open chain with N degrees of freedom may be written as:

$$\Gamma = \mathrm{H}(\mathrm{q})\ddot{\mathrm{q}} + \mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) \tag{5.2}$$

where

$$\begin{aligned}
\Gamma &= N \times 1 \text{ joint force vector,} \\
\mathrm{H}(\mathrm{q}) &= N \times N \text{ joint space inertia matrix,} \\
\mathrm{q} &= N \times 1 \text{ vector of joint coordinates,} \\
\dot{\mathrm{q}}, \ddot{\mathrm{q}} &= N \times 1 \text{ vectors of joint rates and accelerations,} \\
\mathrm{C}(\mathrm{q}, \dot{\mathrm{q}}) &= N \times 1 \text{ vector of centripetal and Coriolis force terms, and} \\
\mathrm{G}(\mathrm{q}) &= N \times 1 \text{ vector of gravity forces.}
\end{aligned}$$

When a closed chain is considered, only one term needs to be added:

$$\Gamma = \mathrm{H}(\mathrm{q})\ddot{\mathrm{q}} + \mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) + \mathrm{J}_{\mathrm{end}}^{\mathrm{T}}(\mathrm{q})\mathrm{f} \tag{5.3}$$

[2] The vector space of end effector variables.

[3] The vector space of joint variables (typically rotations of joints).

where $\mathbf{J}^{\mathrm{T}}_{\mathrm{end}}$ is the $6 \times N$ *Jacobian* matrix, expressed in end effector coordinates, and $\mathbf{f}$ is the 6×1 vector of external forces and moments exerted *by* the last link, also expressed in the end effector coordinates.

Several authors consider the dynamic simulation of manipulators, but few consider closed dynamic chains. Most authors concentrate on the solution to the inverse dynamic problem [23, 11, 4] – that is – determining the forces and torques needed to attain a prescribed motion. Few authors use the same method of assigning kinematic parameters, which make the integration of algorithms from different authors into the same computer program difficult.

Vukobratović and Kirćanski [24] derive an algorithm that solves for certain forces necessary in the upper extremities of a biped robot – to maintain balance – when the motion of the legs are prescribed. In another book Vukobratović and Kirćanski [23] derive an efficient algorithm for the computation of the *Joint Space Inertia Matrix, Coriolis effect matrix* and *gravity loading matrix*, but this algorithm is not considered suitable for the simulation of a closed chain, because it does not give the Jacobian matrix as byproduct. Both methods for obtaining the joint accelerations due to the constraint on a closed chain, as derived by Lilly [2, chapter 5], uses this matrix[4]. Because the methods and algorithms by Lilly are the most complete ones found that can be used to simulate both open and closed chains, it was decided to base the algorithm to be used in this thesis on those algorithms. Lilly refers to other works that demonstrate algorithms for the efficient computation of $\mathbf{C}$ and $\mathbf{G}$.

We will now derive a suitable direct dynamics algorithm to be used in this work. It is not the most efficient algorithm known, but is a compromise between understandability/clarity and efficiency. The solution method will first be shown, then the calculation of the constants are discussed.

5.2.1 Additional notation

The following is useful when considering the concept of successively adding links to a manipulator:
The number N_i is the total number of degrees of freedom of an i-link manipulator, while n_i is the number of degrees of freedom for link i only. The $N_i \times 1$ vector $\ddot{\mathbf{q}}_\mathrm{i}$ represents the complete joint acceleration vector for an i-link manipulator. Let $\ddot{\mathbf{q}}^*_\mathrm{i}$ be the $n_i \times 1$ vector of the joint accelerations about and/or along the axes of joint i.

[4] Other approaches for solving closed chain problems makes use of Lagrange multipliers to solve a linear system that includes the dynamic equation of the manipulator and the tip constraints.

Then these vectors are related as follows:

$$\ddot{\mathrm{q}}_{\mathrm{i}} = \begin{bmatrix} \ddot{\mathrm{q}}_{\mathrm{i-1}} \\ \ddot{\mathrm{q}}_{\mathrm{i}}^{*} \end{bmatrix} \tag{5.4}$$

A similar relationship exists for the joint force vector:

$$\tau_i = \begin{bmatrix} \tau_{i-1} \\ \tau_i^{*} \end{bmatrix} \tag{5.5}$$

The $6 \times N_i$ Jacobian matrix, $\mathbf{J_i}$, for an i-link manipulator is related to the $6 \times N_{i-1}$ Jacobian matrix, $\mathbf{J_{i-1}}$, of an $(i-1)$-link manipulator as follows [18]:

$$\mathbf{J_i} = \begin{bmatrix} {}^{\mathrm{i}}\mathrm{X}_{\mathrm{i-1}}\,\mathbf{J_{i-1}} & \vdots & \phi_i \end{bmatrix} \tag{5.6}$$

where ϕ_i represents the free modes of joint i. Note that any Jacobian is expressed in a particular coordinate frame. The convention used here is that $\mathbf{J_i}$ is expressed in the ith coordinate frame. This, together with the use of spatial notation and the joint description, greatly simplifies the evaluation of the Jacobian.

5.2.2 Joint accelerations for open and closed chains

From the equations of motion of a single N degree-of-freedom chain,

$$\mathbf{\Gamma} = \mathbf{H}(\mathrm{q})\ddot{\mathrm{q}} + \mathbf{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathbf{G}(\mathrm{q}) + \mathbf{J}^{\mathbf{T}}_{\mathbf{end}}(\mathrm{q})\mathbf{f} \tag{5.7}$$

where $\mathbf{f}$ is the 6×1 vector of contact forces and moments exerted by the last link, it can be seen that if $\mathbf{f}$ is known, the mechanism may be treated as an open chain – with an additional known input torque or force at each actuator as a result of the contact force vector. Note that here the Jacobian matrix transforms $\mathbf{f}$ to joint space.

The Jacobian matrix for a N-link manipulator, expressed in the end effector coordinates can be found from:

$$\mathbf{J_{end}} = {}^{\mathrm{end}}\mathbf{X_N}\mathbf{J_N} \tag{5.8}$$

where ${}^{\mathbf{end}}\mathbf{X_N}$ is the transformation from the coordinate frame of the last link to the end effector frame, and $\mathbf{J_N}$ has been calculated recursively using equation 5.6.

The joint acceleration vector can be partitioned into the difference of two terms. The first term corresponds to the chain in the open configuration (contact force removed) while the second term is a function of the contact force vector. This makes it *possible to find a partial solution for the closed-chain joint accelerations before the* *contact force is explicitly known.*

The partitioning may be done with the use of equation 5.3. Solving for the vector of joint accelerations of the closed chain,

$$\begin{aligned} \ddot{\mathrm{q}} &= \mathrm{H}^{-1}(\tau - \mathrm{C}\dot{\mathrm{q}} - \mathrm{G} - \mathrm{J}_{\mathrm{end}}^{\mathrm{T}}\mathrm{f}) && (5.9) \\ &= \mathrm{H}^{-1}(\tau - \mathrm{C}\dot{\mathrm{q}} - \mathrm{G}) - (\mathrm{H}^{-1}\mathrm{J}_{\mathrm{end}}^{\mathrm{T}})\mathrm{f} && (5.10) \\ &= \ddot{\mathrm{q}}_{\mathrm{open}} - \ddot{\mathrm{q}}_{\mathrm{constrained}} && (5.11) \end{aligned}$$

It can also be shown [2, chapter 4] that

$$(\mathrm{H}^{-1}\mathrm{J}_{\mathrm{end}}^{\mathrm{T}}) = \Omega \qquad (5.12)$$

relates the joint acceleration vector and the operational force vector in the absence of bias terms. With this, equation 5.10 can be rewritten:

$$\ddot{\mathrm{q}} = \ddot{\mathrm{q}}_{\mathrm{open}} - \Omega\mathrm{f} \qquad (5.13)$$

5.2.3 End effector acceleration

One can solve explicitly for the end effector acceleration from equation 5.1, but this will involve the calculation of the vector of centripetal and Coriolis force, $\mu(\mathrm{x}, \dot{\mathrm{x}})$, and the vector of gravity forces, $\mathrm{p}(\mathrm{x})$, for the end effector formulation. To avoid this, it is desirable to find a relationship between the joint accelerations, as found through the joint space formulation, and the acceleration of the end effector. It will later be shown that it is necessary to know the inverse operational space inertia matrix, $\mathbf{\Lambda}^{-1}$, in order to find the force on the tip of the last link due to a constraint. A method to calculate this from the inverse joint space inertia matrix will be developed later on.

The relationship between joint accelerations and end effector acceleration follows from the definition of the Jacobian matrix. Using the Jacobian, the end effector velocity is:

$$\dot{\mathrm{x}} = \mathrm{J}_{\mathrm{end}}\dot{\mathrm{q}} \qquad (5.14)$$

Then its time derivative is:

$$\ddot{\mathrm{x}} = \mathrm{J}_{\mathrm{end}}\ddot{\mathrm{q}} + \dot{\mathrm{J}}_{\mathrm{end}}\dot{\mathrm{q}} \qquad (5.15)$$

The above with equation 5.10 can be used to partition the end effector acceleration into terms involving the open chain solution and the tip force:

$$\begin{aligned} \ddot{\mathrm{x}} &= \mathrm{J}_{\mathrm{end}}\mathrm{H}^{-1}(\tau - \mathrm{C}\dot{\mathrm{q}} - \mathrm{G}) + \dot{\mathrm{J}}_{\mathrm{end}}\dot{\mathrm{q}} - (\mathrm{J}_{\mathrm{end}}\mathrm{H}^{-1}\mathrm{J}_{\mathrm{end}}^{\mathrm{T}})\mathrm{f} && (5.16) \\ &= \left[\mathrm{J}_{\mathrm{end}}\mathrm{H}^{-1}(\tau - \mathrm{G}) + (\dot{\mathrm{J}}_{\mathrm{end}} - \mathrm{J}_{\mathrm{end}}\mathrm{H}^{-1}\mathrm{C})\dot{\mathrm{q}}\right] && (5.17) \\ &\quad -(\mathrm{J}_{\mathrm{end}}\mathrm{H}^{-1}\mathrm{J}_{\mathrm{end}}^{\mathrm{T}})\mathrm{f} && (5.18) \\ &= \ddot{\mathrm{x}}_{\mathrm{open}} - \ddot{\mathrm{x}}_{\mathrm{constrained}} && (5.19) \end{aligned}$$

It can be shown [2, chapter 4] that

$$\mathbf{J}_{\mathrm{end}}\mathbf{H}^{-1}\mathbf{J}_{\mathrm{end}}^{\mathrm{T}} = \Lambda^{-1} \tag{5.20}$$

where Λ^{-1} is the inverse operational space matrix for a single chain.[5] Then

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_{\mathrm{open}} - \Lambda^{-1}\mathbf{f} \tag{5.21}$$

5.2.4 Tip constraints

The contact occurs between the last link, body N, of the manipulator, and body $N+1$ – the rigid body in the environment. The joint between these is therefore joint $N+1$.

Let us assume that the two dual bases we use to partition the spatial acceleration and force vectors at the tip are the same. Thus,

$\phi_{N+1} =$ vector space of free (unconstrained) directions for the contact, $[6 \times (6 - n_c)]$,

$\phi_{N+1}^c =$ vector space of constrained directions for the contact, $[6 \times n_c]$,

where n_c is the number of degrees of constraint at the tip.

As was proved earlier, the two vector spaces defined for the contact are orthogonal.

Now the end effector acceleration and general contact force vector can be resolved in the two orthogonal vector spaces of the contact:

$$\ddot{\mathbf{x}} = \phi_{N+1}\mathbf{g} + \phi_{N+1}^c\mathbf{g}^{\mathrm{c}} \tag{5.22}$$

$$\mathbf{f} = \phi_{N+1}\mathbf{h} + \phi_{N+1}^c\mathbf{h}^{\mathrm{c}} \tag{5.23}$$

where $\mathbf{g}$ and $\mathbf{h}$ are the $(6 - n_c) \times 1$ vectors of unconstrained acceleration and force respectively, and $\mathbf{g}^{\mathrm{c}}$ and $\mathbf{h}^{\mathrm{c}}$ are the $n_c \times 1$ constraint acceleration and force vectors respectively. $\mathbf{f}$ is literally $\mathbf{f}_{\mathbf{N+1}}$ and $\mathbf{h}$ and $\mathbf{h}^{\mathrm{c}}$ correspond to τ_{N+1}^{*} and τ_{N+1}^{c} respectively.

For every kind of contact, a unique ϕ_{N+1} and ϕ_{N+1}^c can be specified. Also, for every unknown component of the contact force vector, there is a corresponding known value of the relative angular or linear acceleration of the end effector in the same direction. This is the result of the constraint. From the known components of acceleration and contact force vector, one must find the unknown force components. To facilitate this, contacts can be divided into two classes with reference to their properties:

[5] This will be proven later.

1. Class I contact
   - g^c and h are completely specified,
   - g and h^c are unknown.
2. Class II contact
   - g^c is completely specified,
   - g and h^c are unknown,
   - h is linear function of h^c.

For the problem at hand, we have a Class I contact. Note that body $N+1$ need not be stationary, its motion must just be completely specified.

Calculation of the unknown contact forces (Class I contact)

The dynamic equations of motion expressed in the operational space can be combined with the tip contact model to solve for the unknown force components $\mathbf{h}^c$.

g^c and h are known. From equations 5.21 and 5.22:

$$\begin{aligned} \ddot{\mathrm{x}} &= \ddot{\mathrm{x}}_{\mathrm{open}} - \Lambda^{-1}\mathrm{f} && (5.24) \\ &= \phi_{N+1}\mathrm{g} + \phi^c_{N+1}\mathrm{g}^c && (5.25) \end{aligned}$$

This can be combined with equation 5.23 to give:

$$\phi_{N+1}\mathrm{g} + \phi^c_{N+1}\mathrm{g}^c = \ddot{\mathrm{x}}_{\mathrm{open}} - \Lambda^{-1}(\phi_{N+1}\mathrm{h} + \phi^c_{N+1}\mathrm{h}^c) \qquad (5.26)$$

But, $(\phi^c)^T(\phi^c) = \mathbf{1}$ and $(\phi^c)^T(\phi) = \mathbf{0}$. The unknown quantity g can be eliminated from equation 5.26 by pre-multiplying with $(\phi^c)^T$:

$$\mathrm{g}^c = (\phi^c_{N+1})^T\ddot{\mathrm{x}}_{\mathrm{open}} - (\phi^c_{N+1})^T\Lambda^{-1}(\phi_{N+1}\mathrm{h} + \phi^c_{N+1}\mathrm{h}^c) \qquad (5.27)$$

The only unknown in the above equation is h^c. A solution for h^c can be found from the following linear system:

$$\left[(\phi^c_{N+1})^T\Lambda^{-1}(\phi^c_{N+1})\right]\mathrm{h}^c = \left[(\phi^c_{N+1})^T\ddot{\mathrm{x}}_{\mathrm{open}} - (\phi^c_{N+1})^T\Lambda^{-1}\phi_{N+1}\mathrm{h} - \mathrm{g}^c\right] \qquad (5.28)$$

An explicit solution for it can be found:

$$\mathrm{h}^c = \left[(\phi^c_{N+1})^T\Lambda^{-1}(\phi^c_{N+1})\right]^{-1}\left[(\phi^c_{N+1})^T\ddot{\mathrm{x}}_{\mathrm{open}} - (\phi^c_{N+1})^T\Lambda^{-1}\phi_{N+1}\mathrm{h} - \mathrm{g}^c\right] \qquad (5.29)$$

However, this solution can only be found if $(\phi^c_{N+1})^T\Lambda^{-1}(\phi^c_{N+1})$ is not singular.. This matrix product must therefore have full column rank. In order to fulfill this requirement, the column space of ϕ^c_{N+1} must be a subspace of the column space of $\mathbf{J}_{\text{end}}$, since $\mathbf{\Lambda}^{-1} = \mathbf{J}_{\text{end}}\mathbf{H}^{-1}\mathbf{J}^{\mathbf{T}}_{\text{end}}$ and $\mathbf{H}$ has full column rank. This implies that one should not constrain the end of a manipulator from movements that are already constrained by the joints of the manipulator if the exact solution is sought. This might not always be possible, and the solution procedure for such cases is discussed later.

If $\mathbf{h}^{\mathbf{c}}$ is known then $\mathbf{f}$ is completely specified:

$$\mathrm{f} = \phi_{N+1}\mathrm{h} + \phi^c_{N+1}\mathrm{h}^c \tag{5.30}$$

5.2.5 Calculation of the joint space inertia matrix H

The way in which the Lagrange-Euler or Newton-Euler equations are manipulated into the form of equation. 5.3 results in certain algorithms for the calculation of **H**, **C** and **G**. Not all the resulting algorithms are computationally equally efficient. Some of the more efficient algorithms [2] for the computation of **H** are the:

- Structurally Recursive Method [2],
- Inertia Projection Method [2],
- Modified Composite-Rigid-Body Method [2, 25, 20] and
- Spatial Composite-Rigid-Body Method [2].

Because of its simplicity, the *Structurally Recursive Method* will be used. Its derivation is based on the concept of successively adding individual links to the free end of a serial chain. The forces acting on the added link can be projected onto the motion space of the augmented manipulator, which eliminates some unknown forces. The derivation can be found in Lilly [2, p. 23-27] and will briefly be discussed here. In the derivation that follows, it will be assumed that all joint velocities are zero and that there is no gravity acting on the manipulator, in order to prevent bias terms from entering into the equation, and thus making the identification of the joint space inertia matrix easier. Consider a manipulator with only one link. The free body force equation for this single link can be written as:

$$\mathrm{I}_1\mathrm{a}_1 = \mathrm{f}_1 - {}^2\mathrm{X}_1^{\mathrm{T}}\mathrm{f}_2 \tag{5.31}$$

where $\mathbf{a_1}$ is the 6×1 spatial acceleration vector of link 1 (recall that it includes both linear and angular acceleration), $\mathbf{f_1}$ is the spatial force (moments and linear forces) applied by the base to link 1, and $\mathbf{f_2}$ is the spatial force applied by link 1 to the next link (or any body it contacts) at the origin of coordinate frame 2, and measured in it. The inertia matrix $\mathbf{I}_1$ has been defined earlier. These terms are quantities in Cartesian space. If the above equation is transformed to joint space, the joint space inertia matrix can be identified.

In order to transform the last equation to joint space, each term can be projected onto the motion space of the manipulator by taking the dot product of each term with column vectors that form a base for the motion space of the manipulator. The columns of the Jacobian matrix are basis vectors for the motion space, and the projection can thus be performed by pre-multiplying each term with the transpose of the Jacobian matrix of the single link manipulator:

$$\mathrm{J}_1^T \mathrm{I}_1 \mathrm{a}_1 = \mathrm{J}_1^T \mathrm{f}_1 - \mathrm{J}_1^T \, {}^2\mathrm{X}_1^T \mathrm{f}_2 \tag{5.32}$$

With this operation, the unknown constraint forces and moments at joint 1 has been eliminated, and the projection of f_1 onto the motion space of the first joint is simply τ_1. Because all joint velocities are zero, $\mathrm{a}_1 = \mathrm{J}_1 \ddot{\mathrm{q}}_1$. Now equation 5.32 becomes:

$$(\mathrm{J}_1^T \mathrm{I}_1 \mathrm{J}_1)\ddot{\mathrm{q}} = \tau_1 - \mathrm{J}_{-1}^T \, {}^2\mathrm{X}_1^T \mathrm{f}_2 \tag{5.33}$$

By comparing this with equation 5.3 it can be seen that the joint space inertia matrix of a single link manipulator is:

$$\mathrm{H}_1 = (\mathrm{J}_1^T \mathrm{I}_1 \mathrm{J}_1) \tag{5.34}$$

and the dynamic equation can be written as:

$$\mathrm{H}_1 \ddot{\mathrm{q}} = \tau_1 - \mathrm{J}_{-1}^T \, {}^2\mathrm{X}_1^T \mathrm{f}_2 \tag{5.35}$$

Now consider an $(i-1)$ link manipulator for which the joint space inertia matrix is known. Add a link to the free end to form an i link manipulator. We will now try to establish the relationship between the known $\mathbf{H}_{i-1}$ and $\mathbf{H}_i$. This will lead to a recursion for determining $\mathbf{H}$ for the whole manipulator. The dynamic equation for the $(i-1)$ link manipulator is given as:

$$\mathrm{H}_{i-1} \ddot{\mathrm{q}}_{i-1} = \tau_{i-1} - \mathrm{J}_{i-1}^T \, {}^i\mathrm{X}_{i-1}^T \mathrm{f}_i \tag{5.36}$$

The free-body force equation for the ith link can be written as:

$$\mathrm{I}_i \mathrm{a}_i = \mathrm{f}_i - \, {}^{i+1}\mathrm{X}_i^T \mathrm{f}_{i+1} \tag{5.37}$$

The i-link Jacobian matrix can be found from the $(i-1)$-link Jacobian matrix as:

$$\mathrm{J}_i = \left[{}^i\mathrm{X}_{i-1} \mathrm{J}_{i-1} \vdots \phi_i \right] \tag{5.38}$$

This i-link Jacobian can now be used to project the force terms onto the motion space of the augmented manipulator:

$$\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{I}_{\mathrm{i}}\mathbf{a}_{\mathrm{i}} = \mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} - \mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\, {}^{\mathrm{i+1}}\mathbf{X}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i+1}} \tag{5.39}$$

The projection of the force $\mathbf{f}_{\mathrm{i}}$ onto the motion space of the manipulator can be written in terms of the projection on the motion space of link i, and the projection on the motion space of the first $(i-1)$ links by using equation 5.38:

$$\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} = \begin{bmatrix} \mathbf{J}_{\mathrm{i-1}}^{\mathrm{T}}\, {}^{\mathrm{i}}\mathbf{X}_{\mathrm{i-1}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} \\ \phi_i^T\mathbf{f}_{\mathrm{i}} \end{bmatrix} \tag{5.40}$$

With the use of equation 5.36 and

$$\phi_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} = \tau_{\mathrm{i}}^{*} \tag{5.41}$$

the expression for the projection onto the motion space becomes:

$$\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} = \begin{bmatrix} -\mathbf{H}_{\mathrm{i-1}}\ddot{\mathrm{q}}_{\mathrm{i-1}} + \tau_{\mathrm{i-1}} \\ \tau_i^{*} \end{bmatrix} \tag{5.42}$$

The dimensions of $\mathbf{H}_{\mathrm{i-1}}$ is $N_{i-1} \times N_{i-1}$ where N_{i-1} is the number of degrees of freedom of the $(i-1)$ link manipulator. Let us define the $N_i \times N_i$ matrix $\mathbf{H}_{\mathrm{i-1}}^{*}$ as:

$$\mathbf{H}_{\mathrm{i-1}}^{*} = \begin{bmatrix} \mathbf{H}_{\mathrm{i-1}} & 0 \\ 0 & 0 \end{bmatrix} \tag{5.43}$$

Then,

$$\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i}} = -\mathbf{H}_{\mathrm{i-1}}^{*}\ddot{\mathrm{q}}_{\mathrm{i}} + \tau_{\mathrm{i}} \tag{5.44}$$

Equation 5.39 can be combined with equation 5.44, and the relationship $\mathbf{a}_{\mathrm{i}} = \mathbf{J}_{\mathrm{i}}\ddot{\mathrm{q}}_{\mathrm{i}}$ used to rewrite the projection of the force equation as:

$$(\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{I}_{\mathrm{i}}\mathbf{J}_{\mathrm{i}} + \mathbf{H}_{\mathrm{i-1}}^{*})\ddot{\mathrm{q}}_{\mathrm{i}} = \tau_{\mathrm{i}} - \mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\, {}^{\mathrm{i+1}}\mathbf{X}_{\mathrm{i}}^{\mathrm{T}}\mathbf{f}_{\mathrm{i+1}} \tag{5.45}$$

From this, the joint space inertia matrix is identified as:

$$\mathbf{H}_{\mathrm{i}} = (\mathbf{J}_{\mathrm{i}}^{\mathrm{T}}\mathbf{I}_{\mathrm{i}}\mathbf{J}_{\mathrm{i}} + \mathbf{H}_{\mathrm{i-1}}^{*}). \tag{5.46}$$

This is the basic recursion, and can be used with the definition of $\mathbf{H}_{\mathrm{i-1}}^{*}$ and the fact that $\mathbf{H}_0$ is an empty matrix, to determine the joint space inertia matrix of a complex manipulator recursively.

5.2.6 Calculation of the operational space inertia matrix Λ^{-1}

The inertia matrices are not dependent on whether the manipulator is a closed or open chain. From equation 5.2 the joint accelerations can be expressed as:

$$\ddot{q} = H^{-1}(\Gamma - C\dot{q} - G) \tag{5.47}$$

The generalized force vector, $\mathbf{\Gamma}$, is the projection of the operational force vector, $\mathbf{F}$, onto the motion space of the manipulator, and is given by:

$$\Gamma = J^{T}_{end}(q)F \tag{5.48}$$

This can be substituted into equation 5.47:

$$\begin{aligned} \ddot{q} &= (H^{-1}J^{T}_{end})F - H^{-1}(C\dot{q} + G) && (5.49)\\ &= \Omega F - B_{\ddot{q}}(q, \dot{q}) && (5.50) \end{aligned}$$

where

$$\Omega = (H^{-1}J^{T}_{end}) \tag{5.51}$$

relates the joint acceleration vector and the operational force vector in the absence of bias terms, and where $B_{\ddot{q}}(q, \dot{q})$ is the joint space bias vector, which is only a function of the present state.

From the definition of the Jacobian, the acceleration of the end effector is:

$$\ddot{x} = J_{end}(q)\ddot{q} + \dot{J}_{end}\dot{q} \tag{5.52}$$

When this is combined with equation 5.49, the spatial end effector acceleration may be expressed as:

$$\begin{aligned} \ddot{x} &= (J_{end}H^{-1}J^{T}_{end})F - J_{end}H^{-1}(C\dot{q} + G) + \dot{J}_{end}\dot{q} && (5.53)\\ &= (J_{end}H^{-1}J^{T}_{end})F - B_{\ddot{x}}(x, \dot{x}) && (5.54) \end{aligned}$$

where $B_{\ddot{x}}(x, \dot{x})$ is a bias vector that is also only a function of the present state. Now one can solve for $\mathbf{F}$:

$$\begin{aligned} F &= (J_{end}H^{-1}J^{T}_{end})^{-1}\ddot{x} + (J_{end}H^{-1}J^{T}_{end})^{-1}B_{\ddot{x}}(x, \dot{x}) && (5.55)\\ &= (J_{end}H^{-1}J^{T}_{end})^{-1}\ddot{x} + B_{F}(x, \dot{x}) && (5.56) \end{aligned}$$

where $B_{F}(x, \dot{x})$ represents the bias forces. By comparing the above equation with equation 5.1 the operational space inertia matrix can be identified as:

$$\Lambda(x) = \Lambda(q) = \left[J_{end}(q)H^{-1}J^{T}_{end}(q)\right]^{-1} \tag{5.57}$$

Thus the operational space inertia matrix is a function of position only, and is always a 6 × 6 matrix, irrespective of the number of degrees of freedom, N, of the manipulator. It is very important to note that $\mathbf{\Lambda}$ is only defined if the bracketed term is not singular, thus only if

$$\mathbf{\Lambda}^{-1}(\mathrm{q}) = \left[\mathbf{J}_{\mathrm{end}}(\mathrm{q})\mathbf{H}^{-1}\mathbf{J}_{\mathrm{end}}^{\mathrm{T}}(\mathrm{q})\right] \tag{5.58}$$

has a rank of 6. Since $\mathbf{H}$ has a rank of N (because it is symmetric and positive definite) and is thus always invertible, while $\mathbf{J}$ is a $6 \times N$ matrix for which the rank varies with the position of the manipulator, the rank of the above product depends solely on the Jacobian. If the manipulator has less than six degrees of freedom, $\mathbf{\Lambda}^{-1}$ is always singular. If the manipulator has six or more degrees of freedom and the columns of the Jacobian span $\mathbf{R}^6$, the rank of $\mathbf{J}$ and the above product will be six and the inverse (and thus $\mathbf{\Lambda}$) will be defined. A sufficient condition for $\mathbf{\Lambda}^{-1}$ to be invertible is thus that the Jacobian must have a full column rank of six. If the column rank of the Jacobian is less than six, there are directions in which the tip of the manipulator can not move – because of its structure – and the operational space inertia in these directions are infinite. From this it can be concluded that the inverse operational space inertia matrix, $\mathbf{\Lambda}^{-1}$, is always defined, while the operational space inertia matrix, $\mathbf{\Lambda}$ is only defined explicitly for manipulators with 6 or more degrees of freedom.

5.2.7 Computation of the gravity load vector G, and Coriolis and centripetal force vector C

Both $\mathbf{G}$ and $\mathbf{C}$ can be calculated explicitly. Methods have been found in Paul [4], Zomaya [11] and Vukobratović [23], but these cannot be implemented directly. These methods are also rather complex. An easier and very efficient way to calculate the terms due to gravity and Coriolis and centripetal effects is to use the inverse dynamics algorithm: with the joint accelerations set to zero, but with the joint velocities and positions as they are at that moment, the calculated joint force vector is exactly equal to the forces due to the effect under consideration.

$$\mathbf{\Gamma} = \mathrm{H}(\mathrm{x})\ddot{\mathrm{q}} + \mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) \tag{5.59}$$

but

$$\ddot{\mathrm{q}} = 0 \tag{5.60}$$

therefore

$$\mathbf{\Gamma} = \mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathrm{G}(\mathrm{q}) \tag{5.61}$$

5.2.8 Calculation of the time derivative of the Jacobian matrix

From the foregoing it should be clear that the first time derivative of the Jacobian matrix is needed if one wants to compute the acceleration of the end effector from the joint accelerations. The acceleration of the end effector can be solved for from the operational space dynamic equation (equation 5.1) as well, but this would introduce a great deal of extra computations. Apart from the fact that another linear system has to be solved, the equivalent end effector bias forces also has to be calculated. In order to do this from the joint space bias forces, the time derivative of the Jacobian is needed in any case. To see this, one simply has to examine equation 5.53. From this equation the relationship between the operational space bias vectors and the joint space bias vectors is:

$$\mu(\mathrm{x}, \dot{\mathrm{x}}) + p(\mathrm{x}) = \mathrm{J}\mathrm{H}^{-1}(\mathrm{C}\dot{\mathrm{q}} + \mathrm{G}) + \dot{\mathrm{J}}\dot{\mathrm{q}}. \tag{5.62}$$

Thus, unless we want to calculate the operational bias force vectors from scratch and then solve a linear system with six unknowns, we need to know the time derivative of the Jacobian matrix. In a simulation a simple approximation would be to calculate it from the previous and current Jacobian:

$$\dot{\mathrm{J}} \approx \frac{\mathrm{J}_t - \mathrm{J}_{t-1}}{\delta t}. \tag{5.63}$$

Unless small simulation steps or other special techniques are used, this might not be accurate enough. Let us attempt to calculate the time derivative of the Jacobian from its recursive definition:

$$\mathrm{J}_\mathrm{i} = \left[{}^\mathrm{i}\mathrm{X}_{\mathrm{i}-1}\mathrm{J}_{\mathrm{i}-1} \vdots \phi_i \right]. \tag{5.64}$$

The time derivative of the above equation yields a recursive definition of the time derivative of the Jacobian:

$$\dot{\mathrm{J}}_\mathrm{i} = \left[{}^\mathrm{i}\dot{\mathrm{X}}_{\mathrm{i}-1}\mathrm{J}_{\mathrm{i}-1} + {}^\mathrm{i}\mathrm{X}_{\mathrm{i}-1}\dot{\mathrm{J}}_{\mathrm{i}-1} \vdots \dot{\phi}_i \right] \tag{5.65}$$

By examining the definition of the spatial transformation, its time derivative follows:

$${}^\mathrm{i}\dot{\mathrm{X}}_{\mathrm{i}-1} = \begin{bmatrix} {}^i\dot{\mathrm{A}}_{i-1} & 0 \\ {}^i\dot{\mathrm{A}}_{i-1}\tilde{\mathrm{b}}_i^T +^i \mathrm{A}_{i-1}\dot{\tilde{\mathrm{b}}}_i^T & {}^i\dot{\mathrm{A}}_{i-1} \end{bmatrix}. \tag{5.66}$$

To calculate the time derivative of the rotation transformation might seem to be a problem, but by using the results of Chapter 3 it can easily be calculated: the rotation transformation is in fact a orthonormal second order tensor, and as such its time derivative is given by:

$$ {}^{i}\dot{\mathbf{A}}_{i-1} = {}^{i}\mathbf{A}_{i-1}^{T}\tilde{\omega}_{i}^{relative}\,{}^{i}\mathbf{A}_{i-1}. \tag{5.67} $$

The time derivative of the tensor dual of the relative position vector is trivial to calculate: it is the tensor dual of the relative linear velocity of the two frames, which is only a function of the prismatic joint velocity of joint i. The time derivative of the spatial transformation can thus be calculated. The only other unknown quantity in equation 5.65 is the time derivative of the joint space, ϕ_i.

For all joints with only one rotational degree of freedom, ϕ_i is constant and its derivative is thus zero. For joints with two or more rotational degrees of freedom, the joint space is a function of the joint velocity, and is not constant. The calculation methods when the SDOF notation is used will now be discussed. As the current joint space is calculated in parts (see Section 2.6.4), its time derivative will also be calculated in parts. As discussed earlier, the part of the joint space that corresponds to the first rotation of the gimbal, is given by:

$$ \mathbf{Rot}(z,\theta_z)\,\mathbf{Rot}(y,\theta_y)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{5.68} $$

and the time derivative is simply:

$$ \tilde{\omega}\mathbf{Rot}(z,\theta_z)\,\mathbf{Rot}(y,\theta_y)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5.69} $$

Similarly, the part of the joint space that correspond to the second rotation of the gimbal is given by:

$$ \mathbf{Rot}(z,\theta_z)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{5.70} $$

and its time derivative is simply:

$$ \tilde{\omega}\mathbf{Rot}(z,\theta_z)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{5.71} $$

In the above, ω is the part of the angular velocity due to the rotations around the z- and y-axes, or the z-axis, respectively. The part of the joint space that corresponds to the third rotation of the gimbal is not a function of the joint positions of joint i

and as such the part of the time derivative of the joint space that corresponds to it is zero. The prismatic joints have no contribution to the time derivative of the joint space.

After considering the above results, one may well decide to live with the error of using the approximation to the time derivative. However, the accuracy of the time derivative used is equal in importance to the accuracy of the Jacobian itself. A simple test donr on the system described later in this section quickly showed this. The magnitude of the contributions of the terms containing the Jacobian and the time derivative of the Jacobian respectively towards the open chain operational space acceleration (equation 5.52) is shown in Figure 5.1. In this figure the parts of the acceleration due to each of the terms are of the same order. Despite this, the numerical approximation was used in the implementation of the algorithms, and the performance proved to be very satisfactory.

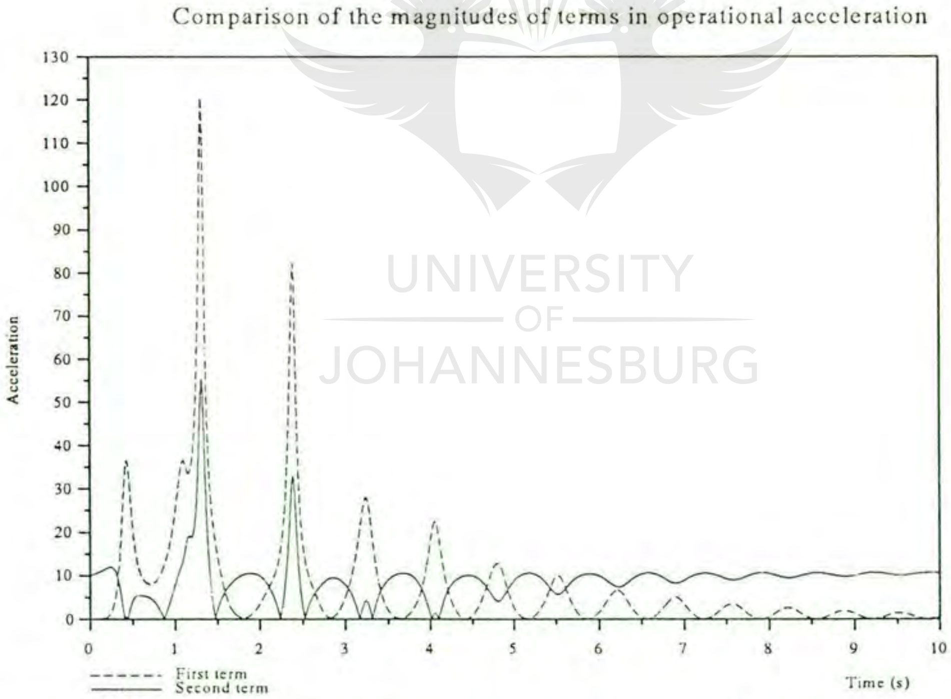


Figure 5.1: The magnitudes of the two terms in equation 5.52 during the simulation of the four link mechanism described later in this chapter.

5.3 Simulation algorithm

To simulate the dynamic behavior of the robot. the following algorithm can·be used:

1. Given:
    - ϕ_i and $\mathbf{I_i}$ for $i = 1, \ldots, N$
    - ϕ_{N+1}
    - h and g^c
    - $\mathbf{J_0}$ is empty
    - $\mathbf{H_0}$ is empty
    - The force vector, τ.
2. Calculate spatial transformations ${}^{\mathrm{i}}\mathrm{X_{i-1}}$ for $i = 1, \ldots, N$
3. Calculate $\mathbf{H}$, $\mathbf{G}$ and $\mathbf{C}$
    - $\mathbf{J_i} = \left[{}^{\mathrm{i}}\mathrm{X_{i-1}}\mathbf{J_{i-1}} \vdots \phi_i\right]$
    - $\mathbf{H_i} = (\mathbf{J_i^T}\mathbf{I_i}\mathbf{J_i} + \mathbf{H_{i-1}^*})$
    - $\mathbf{\Gamma} = \mathbf{C}(\mathrm{q}, \dot{\mathrm{q}})\dot{\mathrm{q}} + \mathbf{G}(\mathrm{q})$ as calculated with the inverse dynamics algorithm
4. Calculate $\mathbf{\Omega}$ and $\mathbf{\Lambda^{-1}}$
    - $\mathbf{J_{end}} = {}^{\mathrm{end}}\mathrm{X_N}\,\mathbf{J_N}$
    - Calculate $\dot{\mathbf{J}}_{\mathrm{end}}$
    - $\mathbf{\Omega} = \mathbf{H^{-1}J_{end}^T}$
    - $\mathbf{\Lambda^{-1}} = \mathbf{J_{end}\Omega}$
5. Open chain solution
    - $\ddot{\mathrm{q}}_{\mathrm{open}} = \mathbf{H^{-1}}(\tau - \mathbf{C}\dot{\mathrm{q}} - \mathbf{G})$
6. Calculate the unknown contact force vector
    - $\ddot{\mathrm{x}}_{\mathrm{open}} = \mathbf{J_{end}}\ddot{\mathrm{q}}_{\mathrm{open}} + \dot{\mathbf{J}}_{\mathrm{end}}\dot{\mathrm{q}}$
    - $\mathrm{h}^c = \left[(\phi^c_{N+1})^T \mathbf{\Lambda^{-1}} (\phi^c_{N+1})\right]^{-1} \left[(\phi^c_{N+1})^T \ddot{\mathrm{x}}_{\mathrm{open}} - (\phi^c_{N+1})^T \mathbf{\Lambda^{-1}} \phi_{N+1}\mathrm{h} - \mathrm{g}^c\right]$
    - $\mathrm{f} = \phi_{N+1}\mathrm{h} + \phi^c_{N+1}\mathrm{h}^c$
7. Closed chain joint accelerations
    - $\ddot{\mathrm{q}} = \ddot{\mathrm{q}}_{\mathrm{open}} - \mathbf{\Omega}\mathrm{f}$
8. Integrate to the next state.

5.4 Some detail relating to the application of algorithms

5.4.1 Tip constraints

The constraint directions of the interaction of the tip with the environment is expressed in the tip coordinate frame. If this frame is fixed to the last link, it orientation will continually change as the joint positions of a manipulator changes. In this case the tip constraints continually change – even though ϕ^c_{N+1} is constant. This is like having the source of the constraints rotate with the last link. There may be cases where such constraints actually occur, but usually the constraints are exerted by an environment that is fixed relative to the base frame of the system. The constraint directions thus remain constant relative to the base frame. To implement such constraints, the tip frame can be fixed to the last link and the tip constraints transformed to this frame, or the tip frame can have the same orientation as the base frame – with the origin at a fixed point relative to the last link. The second proposal is favoured by the author. In the dynamic algorithms, the spatial transformation from frame N to the tip frame is needed. The rotation transformation from frame N to the tip frame is a function of the orientation of the last link and continually changes. However, this rotation can easily be calculated from the other rotation transformations that have already been calculated. To show this, simply consider that the result of the successive rotation transformations from the base to the tip has to be the identity transformation:

$$ {}^{end}\mathbf{A}_N \, {}^N\mathbf{A}_{N-1} \ldots {}^2\mathbf{A}_1 \, {}^1\mathbf{A}_0 = \mathbf{1} \tag{5.72} $$

From this we find the rotation from frame N to the tip frame (remember that the inverse of a rotation matrix is the transpose of this matrix since it is orthonormal):

$$ \begin{aligned} {}^{end}\mathbf{A}_N &= \mathbf{1} \, {}^1\mathbf{A}_0^T \, {}^2\mathbf{A}_1^T \ldots {}^N\mathbf{A}_{N-1}^T & (5.73) \\ &= {}^N\mathbf{A}_{N-1} \ldots {}^2\mathbf{A}_1 \, {}^1\mathbf{A}_0 & (5.74) \end{aligned} $$

5.4.2 Integration error

One can use sophisticated integration techniques to minimize integration error, but in this kind of system, the integration errors compound abnormally fast (especially for the joint positions) because of the strong coupling between the variables. What happens is that small integration errors cause displacements of the tip of the manipulator from its constraint position. This is not due to an inaccurately calculated tip force, but only integration error. Now the constraint has actually changed and the solutions in the following time steps are calculated for this different system, and not the original system.

To minimize this compounding effect, one can try to restore the system so that the position of the tip constraint remains the same. The difference between the calculated position of the tip and its constraint position is known explicitly. The problem is now how to distribute this displacement error to errors in the respective joint positions. Kinematically this can be a singular problem. A more sensible solution is to make use of the dynamics of the system to distribute the error between the joints in the system. This can be done by adding a small "corrective" force to the tip force at the end of the manipulator. This force can simply be a linear function of the velocity and position discrepancies [2]:

$$\mathbf{f}_{\text{corrective}} = \mathbf{K}_v(\dot{\mathbf{x}}_c - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}), \tag{5.75}$$

where $\dot{\mathbf{x}}_c$ and $\mathbf{x}_c$ are the constrained end effector position and velocity vectors, and $\mathbf{K}_v$ and $\mathbf{K}_p$ are the rate and position feedback gains.

5.5 Worked example

We will now work through a simple example step-by-step for the first time step in the simulation.

5.5.1 Problem definition

The system we will consider is not very complex: it is a single serial chain with four links, with all the joints having only one degree of freedom,[6] as in Figure 5.2. Once again we will use the SDOF kinematic notation since it is so convenient. We will include joint friction in the system. In order that the calculation of the contact force in closed chain mechanisms can be illustrated, let us also constrain the tip of the manipulator (the end of link four).

There are many ways in which one can constrain the acceleration of the tip. We might want to constrain its linear motion to moving in some straight line, or constrain it from rotating etc. Further, the constraint is expressed in the tip frame of the manipulator. If the tip frame is fixed to link four, very interesting constraints can be constructed, such as constraining the motion of the tip in its instantaneous z-direction – which is always changing. More often the motion of the tip is constrained relative to the base. We will constrain our manipulator in this way: the tip

[6] This manipulator happens to be a planar mechanism, but this is immaterial as all the algorithms used are for general 3D mechanisms that can have any finite number of links and multiple degree of freedom joints. Although *kinematic* singularities are more likely to occur in three dimensional structures, these singularities has no influence on any of the dynamics algorithms.

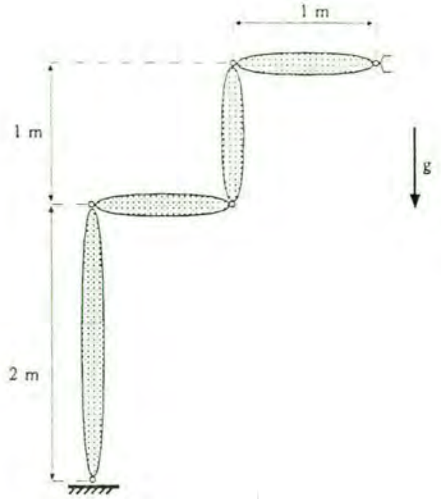


Figure 5.2: A simple serial chain mechanism that is used in the worked example.

is constrained from moving in the base x- and z-directions.

The coordinate frames are assigned as in Figure 5.3. The structure and dynamic properties of the manipulator can now be defined. The joint parameters and joint space are as in Table 5.1. The inertial properties of all the links are given in Table 5.2.

5.5.2 Calculation of spatial inertias

For all the links, the inertia tensor around the link frame origin can be calculated from the inertia tensors around the respective centers of mass by using the parallel axis theorem. For link 1 this is done as follows:

$$
\begin{aligned}
\bar{\mathbf{I}}_1 &= \underline{\mathbf{I}}_1 - m_1\tilde{\mathbf{s}}_1\tilde{\mathbf{s}}_1 && (5.76)\\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0{,}1 \end{bmatrix} - 2\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0{,}1 \end{bmatrix} && (5.77)
\end{aligned}
$$

Now the spatial inertia of link 1 can be calculated:

$$
\mathbf{I}_1 = \begin{bmatrix} \bar{\mathbf{I}}_i & m_1\tilde{\mathbf{s}}_1 \\ m_1\tilde{\mathbf{s}}_1 & m_1\mathbf{1}_{3\times 3} \end{bmatrix}. \qquad (5.78)
$$

Link i	Joint parameters γ_i	Joint space ϕ_i
1	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
3	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
4	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Table 5.1: The SDOF joint parameters and joint spaces of the four link manipulator.

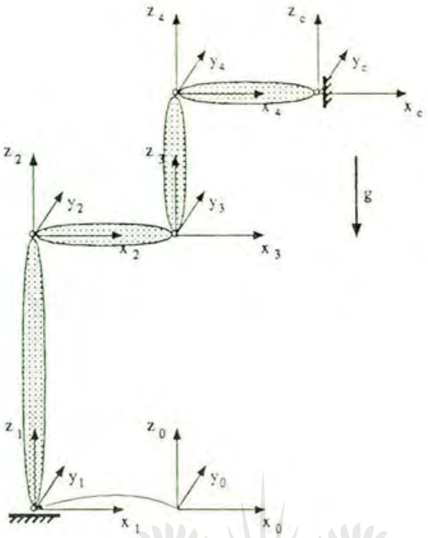


Figure 5.3: The assignment of coordinate frames to the four link manipulator.

Link	Mass	Position of center of mass	Inertia Tensor
1	2	$\begin{bmatrix}0\\0\\1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&0,1\end{bmatrix}$
2	1	$\begin{bmatrix}0,5\\0\\0\end{bmatrix}$	$\begin{bmatrix}0,02&0&0\\0&0,2&0\\0&0&0,2\end{bmatrix}$
3	1	$\begin{bmatrix}0\\0\\0,5\end{bmatrix}$	$\begin{bmatrix}0,2&0&0\\0&0,2&0\\0&0&0,02\end{bmatrix}$
4	1	$\begin{bmatrix}0,5\\0\\0\end{bmatrix}$	$\begin{bmatrix}0,02&0&0\\0&0,2&0\\0&0&0,2\end{bmatrix}$

Table 5.2: The inertial properties of the four link manipulator.

$$= \begin{bmatrix} 3 & 0 & 0 & 0 & -2 & 0 \\ 0 & 3 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0,10 & 0 & 0 & \\ 0 & 2 & 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}. \tag{5.79}$$

The spatial inertias of the other links are as follows:

$$\mathbf{I}_2 = \begin{bmatrix} 0,02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,45 & 0 & 0 & 0 & -0,5 \\ 0 & 0 & 0,45 & 0 & 0,5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 1 & 0 \\ 0 & -0,5 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.80}$$

$$\mathbf{I}_3 = \begin{bmatrix} 0,45 & 0 & 0 & 0 & -0,5 & 0 \\ 0 & 0,45 & 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0,02 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 1 & 0 & 0 \\ -0,5 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.81}$$

$$\mathbf{I}_4 = \begin{bmatrix} 0,02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,45 & 0 & 0 & 0 & -0,5 \\ 0 & 0 & 0,45 & 0 & 0,5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 1 & 0 \\ 0 & -0,5 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.82}$$

The inertial properties of the links have now been completely specified. The initial conditions can now be specified.

5.5.3 Initial conditions

The base is given a linear acceleration of $10\,ms^{-2}$ upward to account for the influence of gravity. The initial joint positions have already been defined implicitly by the joint parameters as:

$$\mathrm{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.83}$$

For this example we also put the initial joint velocities to zero:

$$\dot{\mathrm{q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.84}$$

5.5.4 Repetitive calculations

The following steps all have to be repeated for each time step in the simulation. We start by calculating the bias joint forces - which includes Coriolis, centripetal and friction effects. These forces are calculated by using the inverse dynamics algorithm for the current state and no joint accelerations. The calculation is not shown here, but the result is:

$$\mathbf{\Gamma}_{\text{bias}} = \begin{bmatrix} -30 \\ -30 \\ -5 \\ -5 \end{bmatrix}. \tag{5.85}$$

We are using viscous friction, and at this stage it has no influence since the joint velocities are all zero. We also need to calculate the spatial transformations between the successive links in the manipulator to apply the direct dynamics algorithm. Because of the convenient choice of link frames made possible by the use of the SDOF notation, all the orientation transformations are identity transformations:

$${}^{i}\mathbf{A}_{i-1} = \mathbf{1}_{3\times 3}, \quad \text{for } i = 1, 2, 3, 4. \tag{5.86}$$

The spatial transformations are calculated from the definition of the spatial transformation. The results are:

$${}^{1}\mathbf{X}_{0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.87}$$

$${}^{2}\mathbf{X}_{1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ -2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.88}$$

$$
{}^{1}\mathbf{X}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.89}
$$

$$
{}^{1}\mathbf{X}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.90}
$$

Now we will iteratively calculate the Jacobian and joint space inertia matrix using the *Structurally Recursive Method* [2]. First we consider only link one. The Jacobian is simply the joint space of joint one:

$$
\mathbf{J}_1 = \left[{}^{1}\mathbf{X}_0 \mathbf{J}_0 \vdots \phi_1 \right] \tag{5.91}
$$

$$
= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ } (\mathbf{J_0} \text{ is empty}). \tag{5.92}
$$

The joint space inertia matrix for the single link manipulator is simply the moment of inertia of link one around the link frame y-axis:

$$
\mathbf{H}_1 = (\mathbf{J}_1^{\mathrm{T}} \mathbf{I}_1 \mathbf{J}_1 + \mathbf{H}_0^{*}) \tag{5.93}
$$

$$
= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 & -2 & 0 \\ 0 & 3 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0,10 & 0 & 0 & \\ 0 & 2 & 0 & 2 & 0 & 0 \\ -2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ } (\mathbf{H}_0^{*} \text{ empty})
$$

$$
= [3] \tag{5.94}
$$

Now the augmented chain with two links is considered. The Jacobian for it is given by:

$$
\mathbf{J}_2 = \left[{}^{2}\mathbf{X}_1 \mathbf{J}_1 \vdots \phi_2 \right] \tag{5.95}
$$

$$= \left[\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ -2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \vdots \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right] \tag{5.96}$$

$$= \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.97}$$

The Jacobian has a simple physical interpretation. The meaning of $\mathbf{J}_2$ is that a unit rotation of the first joint causes a unit rotation around the y-axis and two units linear displacement in the x-direction at the second link frame origin. A unit rotation of the second joint only causes a rotation around the y-axis. The joint space inertia of the two link manipulator is:

$$\mathbf{H}_2 = (\mathbf{J}_2^{\mathbf{T}} \mathbf{I}_2 \mathbf{J}_2 + \mathbf{H}_1^*) \tag{5.98}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0{,}02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0{,}45 & 0 & 0 & 0 & -0{,}5 \\ 0 & 0 & 0{,}45 & 0 & 0{,}5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0{,}5 & 0 & 1 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.99}$$

$$= \begin{bmatrix} 7{,}45 & 0{,}45 \\ 0.45 & 0{,}45 \end{bmatrix} \tag{5.100}$$

From this joint space inertia matrix for the two link manipulator, it is clear that the inertia around the joint axis of joint one includes both the inertia of link 1 and the inertia of link 2 transformed to the first joint. This has to be so, since a rotation of joint 1 also displaces link 2 linearly. The above should clarify the calculation of the respective partial Jacobians and joint space inertia matrices sufficiently. The results for three and four links are as follow:

$$\mathbf{J}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 0 \end{bmatrix}, \tag{5.101}$$

$$\mathbf{H}_3 = \begin{bmatrix} 14,9 & 2.9 & 1,45 \\ 2,9 & 1.9 & 0,45 \\ 1,45 & 0.45 & 0,45 \end{bmatrix}, \tag{5.102}$$

$$\mathbf{J}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix}, \tag{5.103}$$

$$\mathbf{H}_4 = \begin{bmatrix} 26.35 & 8.35 & 5,4 & 0.95 \\ 8,35 & 5,35 & 2,4 & 0.95 \\ 5,4 & 2.4 & 1,9 & 0,45 \\ 0,95 & 0.95 & 0,45 & 0,45 \end{bmatrix}. \tag{5.104}$$

We can easily find the Jacobian of the manipulator at its tip by transforming $\mathbf{J}_4$ to the tip frame:

$$\mathbf{J}_{\mathrm{end}} = {}^{\mathrm{end}}\mathbf{X}_4\mathbf{J}_4. \tag{5.105}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & -1 & -1 \end{bmatrix} \tag{5.106}$$

For the other time steps we will need to know the time derivative of the Jacobian, $\dot{\mathbf{J}}_{\mathrm{end}}$, but since the joint velocities are all zero at this stage, it is not needed. It is obvious that $\dot{\mathbf{J}}_{\mathrm{end}}$ is a 6×4 matrix of zeros since the angular velocities of all the links are zero. From the quantities calculated above, it is easy to calculate the inverse operational space inertia matrix:

$$\boldsymbol{\Omega} = \mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}}_{\mathrm{end}} \tag{5.107}$$

$$
= \begin{bmatrix} 26{,}35 & 8{,}35 & 5{,}4 & 0{,}95 \\ 8{,}35 & 5{,}35 & 2{,}4 & 0{,}95 \\ 5.4 & 2{,}4 & 1{,}9 & 0{,}45 \\ 0{,}95 & 0.95 & 0{,}45 & 0{,}45 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 & 0 & -2 \\ 0 & 1 & 0 & 1 & 0 & -2 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} . \tag{5.108}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 & -0.0078431 & 0 & 0 \\ 0 & -0.5865103 & 0 & 0{,}0078431 & 0 & 0{,}0586510 \\ 0 & 0.5865103 & 0 & 0.7058824 & 0 & -0{,}0586510 \\ 0 & 2.8739003 & 0 & -0.7058824 & 0 & -2{,}28739 \end{bmatrix} \tag{5.109}
$$

$$
\Lambda^{-1} = \mathbf{J}_{\mathrm{end}}\boldsymbol{\Omega} \tag{5.110}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -0{,}0078 & 0 & 0 \\ 0 & -0.5865 & 0 & 0{,}0078 & 0 & 0{,}0587 \\ 0 & 0{,}5865 & 0 & 0{,}7059 & 0 & -0{,}0587 \\ 0 & 2{,}8739 & 0 & -0{,}7059 & 0 & -2{,}2874 \end{bmatrix}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2{,}8739003 & 0 & 0 & 0 & -2{,}28739 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}69019610 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2{,}28739 & 0 & 0 & 0 & 2{,}228739 \end{bmatrix} . \tag{5.111}
$$

The open chain joint accelerations and operational space acceleration can now be calculated:

$$
\ddot{\mathbf{q}}_{\mathrm{open}} = \mathbf{H}^{-1}(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathrm{bias}}) \tag{5.112}
$$

$$
= \begin{bmatrix} 0{,}1137 & -0{,}1137 & -0{,}2353 & 0{,}2353 \\ -0{,}1137 & 0{,}6416 & -0{,}2926 & -0{,}8218 \\ -0{,}2353 & -0{,}2926 & 1{,}7043 & -0{,}5900 \\ 0{,}2353 & -0{,}8218 & -0{,}5900 & 4{,}0504 \end{bmatrix} \left(- \begin{bmatrix} -30 \\ -30 \\ -5 \\ -5 \end{bmatrix} \right) \tag{5.113}
$$

$$
= \begin{bmatrix} 0 \\ 10{,}2639296 \\ -10{,}2639296 \\ -0{,}2932551 \end{bmatrix} . \tag{5.114}
$$

The operational space acceleration (expressed in the tip coordinate frame) follows from the joint accelerations:

$$
\ddot{\mathbf{x}}_{\mathrm{open}} = \mathbf{J}_{\mathrm{end}}\ddot{\mathbf{q}}_{\mathrm{open}} + \dot{\mathbf{J}}_{\mathrm{end}}\dot{\mathbf{q}} \tag{5.115}
$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & -2 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 10,2639296 \\ -10,2639296 \\ -0,2932551 \end{bmatrix} \tag{5.116}$$

$$= \begin{bmatrix} 0 \\ -0,29325510 \\ 0 \\ 0 \\ -9,9706745 \end{bmatrix} \tag{5.117}$$

We have now come to the essence of the closed-chain solution: calculating the constraint force on the tip of the manipulator. We want the Cartesian position of the end of the manipulator to stay fixed relative to the base. To implement this constraint, the tip frame has to have a constant orientation relative to the base coordinate frame, for instance having the same orientation as the base coordinate frame. The rotation transformation from the fourth link frame to the tip link frame changes as the manipulator moves, but can easily be calculated – indeed for the current state it is trivially the 3 × 3 identity matrix.

The constraints on the tip of the manipulator should be specified with all the other definitions of the manipulator structure and properties, but it has deliberately been postponed untill now since some finer points of this have to be discussed. In essence, the tip will have only one degree of freedom: rotation around y. One can thus constrain the other five independent motions of the link. However, this is not necessary since the tip of the link cannot move in the y-direction (planar mechanism), nor can it rotate around the x- and z-directions. It is a result of the structure of the manipulator. Constraining these motions as well, leads to redundant constraints on the tip of the manipulator. It will lead to singular inertial matrices in the operational space dynamic equation for the tip motion (equation 5.28), but the system can still be solved.

Although one can avoid this problem, for complex systems it might not be clear from the onset which motions the tip of the manipulator are not capable of, or the manipulator can move into a singular (kinematic) position at some stage – leading to the formation of redundant constraints that did not exist at the onset of the simulation. It is thus essential that the constraint force can be solved for systems with redundant constraints as well. To elaborate on how to handle such a system, the tip of the manipulator will be subjected to the five constraints mentioned above and the solution procedure demonstrated. The motion space and constraint space is:

$$\varphi_{N+1} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.118}$$

$$\varphi^{c}_{N+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.119}$$

The acceleration of the tip in the constrained directions has to be zero, since we want the tip to stay fixed relative to the base:

$$\mathrm{g}^{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.120}$$

We will not apply any forces in the free directions of the tip, thus the moment around y is set to zero:

$$\mathrm{h} = [0] \tag{5.121}$$

In order to determine the forces in the constrained directions at the tip, we have to solve the following linear system:

$$\left[(\phi^{c}_{N+1})^{T}\mathbf{\Lambda}^{-1}(\varphi^{c}_{N+1})\right]\mathrm{h}^{c} = \left[(\phi^{c}_{N+1})^{T}\ddot{\mathrm{x}}_{\mathrm{open}} - (\phi^{c}_{N+1})^{T}\mathbf{\Lambda}^{-1}\varphi_{N+1}\mathrm{h} - \mathrm{g}^{c}\right]. \tag{5.122}$$

The system is in the form $\mathbf{A}\mathrm{x} = \mathbf{b}$. The first term, representing $\mathbf{A}$ is:

$$\begin{aligned} \mathbf{A} &= \left[(\phi^{c}_{N+1})^{T}\mathbf{\Lambda}^{-1}(\varphi_{N+1})\right] \qquad (5.123) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2{,}8739003 & 0 & 0 & 0 & -2{,}28739 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0{,}69019610 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2{,}28739 & 0 & 0 & 0 & 2{,}228739 \end{bmatrix} \end{aligned}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.124}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,6901961 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.228739 \end{bmatrix}. \tag{5.125}$$

A represents the image of the inverse operational space inertia in the constraint directions of the contact. Because three of the constraints specified are redundant, **A** does not have full column rank but has rank equal to two. The right side term of the linear system is:

$$\mathbf{b} = \left[(\phi^c_{N+1})^T \ddot{\mathbf{x}}_{\text{open}} - (\phi^c_{N+1})^T \Lambda^{-1} \phi_{N+1} \mathbf{h} - \mathbf{g}^c\right] \tag{5.126}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -0,29325510 \\ 0 \\ 0 \\ -9,9706745 \end{bmatrix}$$

$$- \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2,8739 & 0 & 0 & 0 & -2,2874 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,6902 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2,2874 & 0 & 0 & 0 & 2,2287 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [0]$$

$$- \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.127}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -9,9706745 \end{bmatrix}. \tag{5.128}$$

This represents the difference between the open chain operational space acceleration and the required closed chain acceleration of the tip. We can now proceed to solving the linear system. We have a system of dimension five, but with only two leading variables (in the third and fifth rows). The other three variables are free variables. The system can be solved by applying Gauss-Jordan elimination on the augmented matrix of the system. The general solution is now the sum of the homogeneous solution and a particular solution [2, 15]. The particular solution depends on the values chosen for the free variables. The free variables represent the components of the tip force in the redundant constraint directions. The magnitudes of these components has no effect on the motion of the manipulator: the inertia of the manipulator in these directions is infinite. In the system of equations above, these components are all multiplied by zero. One can thus make any convenient choice of values for the free variables. A convenient choice would be to let all the free variables be zero. Only the homogeneous solution is thus used. The homogeneous solution to the system is:

$$\mathrm{h}^c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -4,4736842 \end{bmatrix}. \tag{5.129}$$

This means that only an upward force of $4,4736842\,N$ needs to be exerted on the tip of the manipulator to keep it in the current position – to obey the constraints. The force the tip exerts on the environment is thus:

$$\begin{aligned} \mathrm{f} &= \phi_{N+1}\mathrm{h} + \phi^c_{N+1}\mathrm{h}^c \qquad (5.130) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4,4736842 \end{bmatrix}. \qquad (5.131) \end{aligned}$$

At last the closed chain joint accelerations can be determined:

$$\begin{aligned} \ddot{\mathrm{q}} &= \ddot{\mathrm{q}}_{\mathrm{open}} - \Omega\mathrm{f} \qquad (5.132) \\ &= \begin{bmatrix} 0 \\ 10,264 \\ -10,264 \\ -0,293 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & -0,008 & 0 & 0 \\ 0 & -0,587 & 0 & 0,008 & 0 & 0,059 \\ 0 & 0,587 & 0 & 0,706 & 0 & -0,059 \\ 0 & 2,874 & 0 & -0,706 & 0 & -2,287 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4,474 \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} 0 \\ 10.5263158 \\ -10.5263158 \\ -10.5263158 \end{bmatrix} rad/s^2. \tag{5.133}$$

This is an exiting result: apart from link one that does not yet need to accelerate, the other links all have the same magnitudes of joint accelerations, as one would expect from the geometry.

5.5.5 Simulation results

The first ten seconds of the response of this system has been simulated[7], using the algorithms discussed in this chapter and previous ones.

Viscous joint friction has been included. For the simulation, each joint had a friction coefficient of $0,25\,Ns/rad$. Non-adaptive first order Euler integration has been used. This is the simplest method, and the accuracy is not good, especially in this system that has very high acceleration spikes. For the results that follow integration time steps of $0,001\,s$ have been used. The simulated joint accelerations are shown in Figure 5.4.

It can be seen that this simple system has a quite complex response. This is largely due to the inertial effects and bias forces, and the interaction of the links. The joint velocities and positions during the simulation can be seen in Figures 5.5 and 5.6 respectively.

By comparing the previous three figures, it can be seen that as the joint velocities die away, the bias accelerations get smaller and more regular. In the limit, the oscillations will become simple second order responses. The very sharp acceleration peaks make large contributions to the integration errors – showing why ideally adaptive step length integration algorithms should be used. The joint accelerations during the first two and a half seconds of the simulation are shown in Figure 5.7

No matter which integration algorithm is used, the integration error can accumulate[8] – with the effect that the tip of the manipulator moves away from the position it is constrained to. This is not due to inaccurate values for the constraint force, but integration error. To counteract the effect of these integration errors, a corrective

[7] The Scilab executable "four_link_closed.exe" accomplishes this.

[8] Especially for the joint positions.

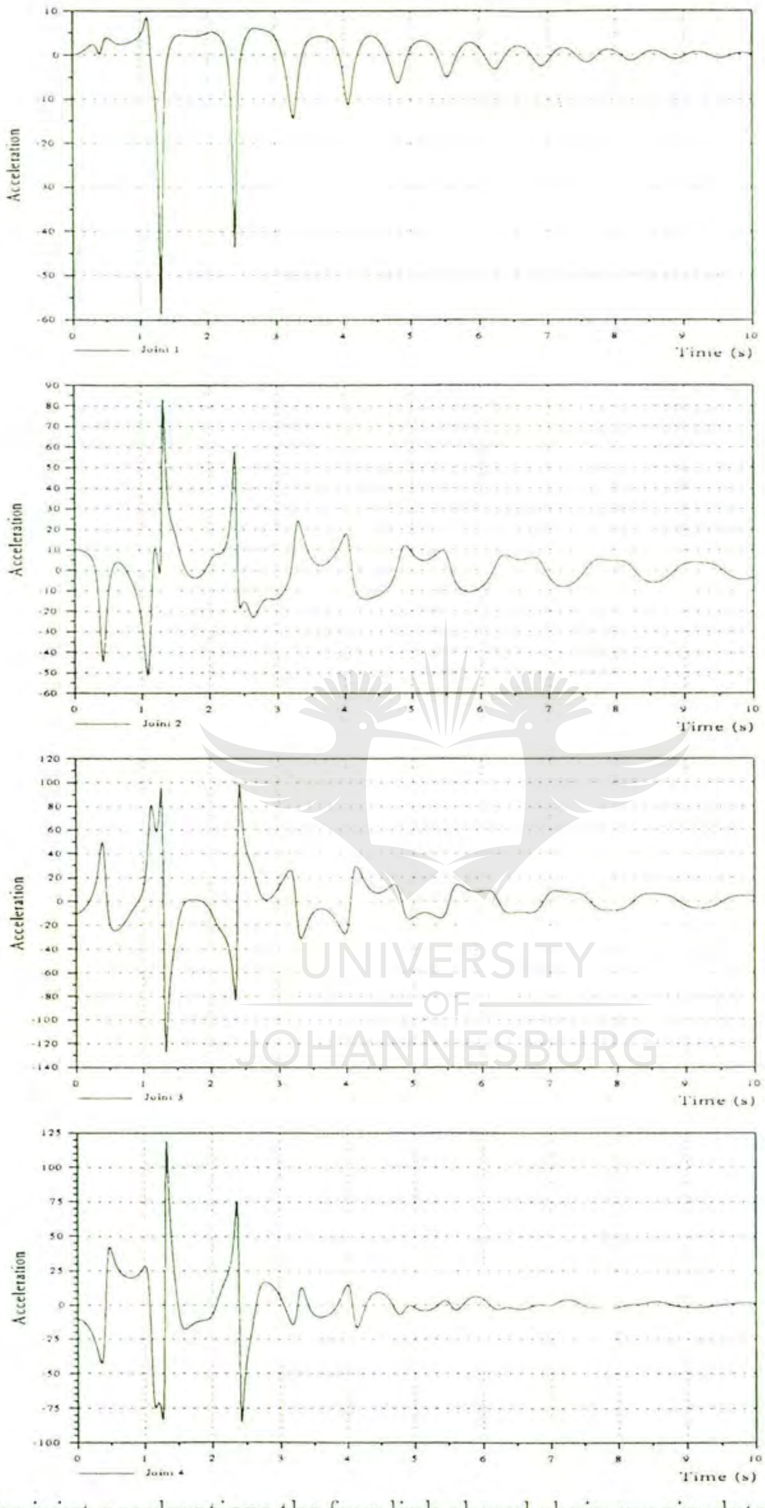


Figure 5.4: The joint accelerations the four link closed chain manipulator experiences under the influence of gravity. As all joints are rotational, the units for the repsective joint accelerations are all rad/s^2.

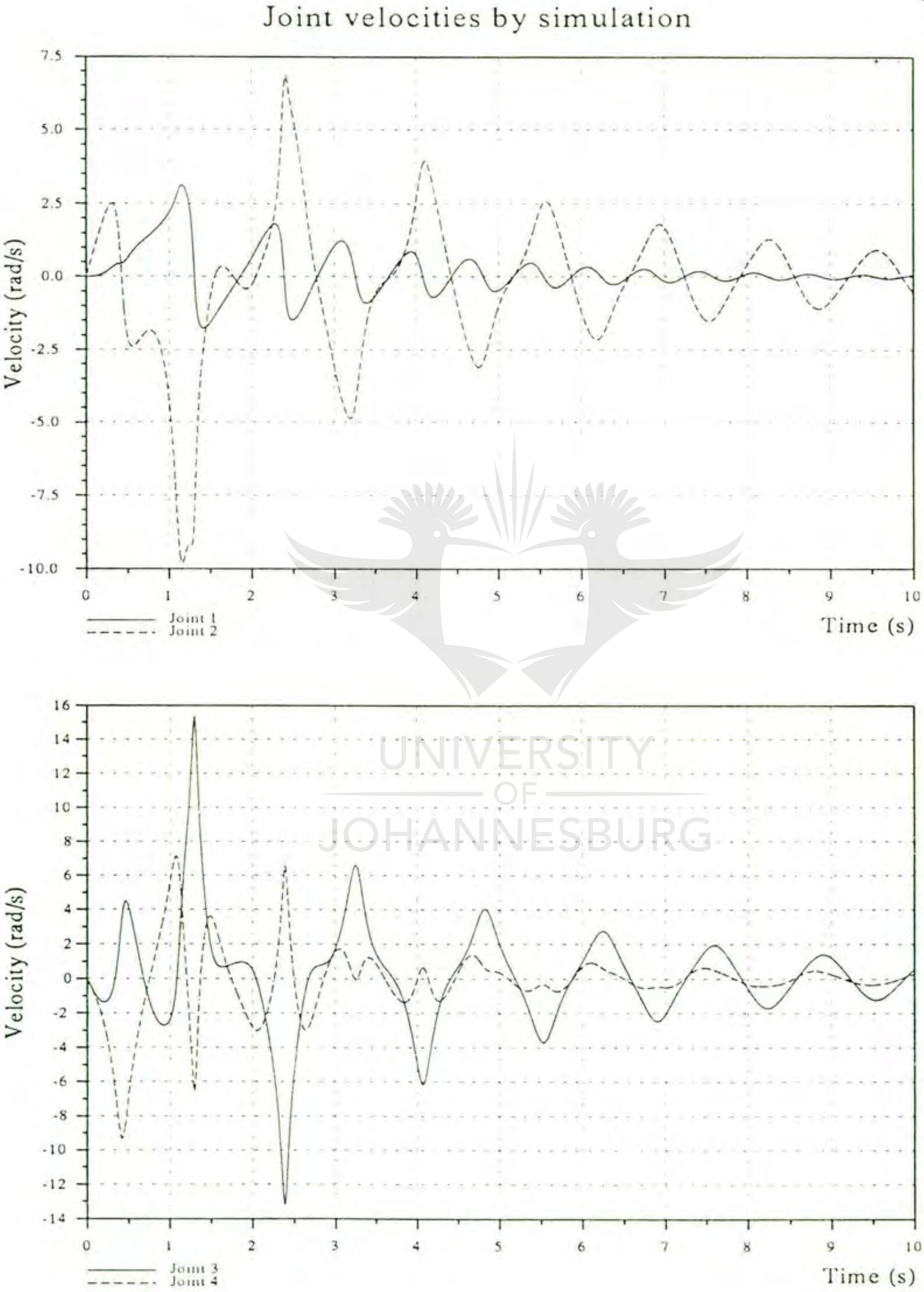


Figure 5.5: The joint velocities of the four link closed chain manipulator under the influence of gravity.

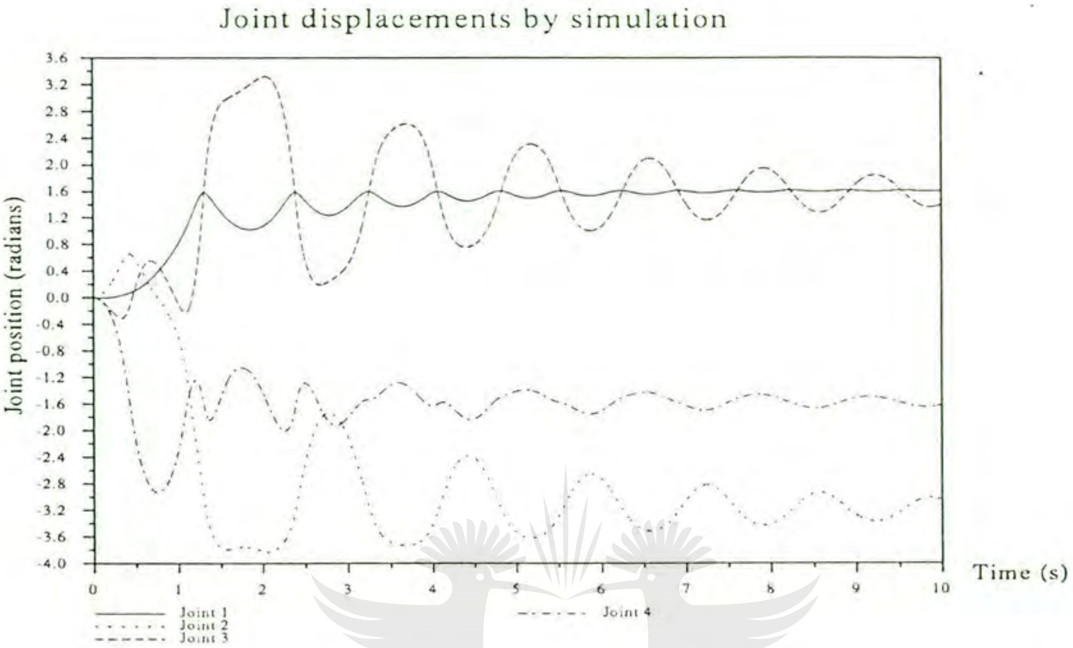


Figure 5.6: The joint positions of the four link closed chain manipulator under the influence of gravity.

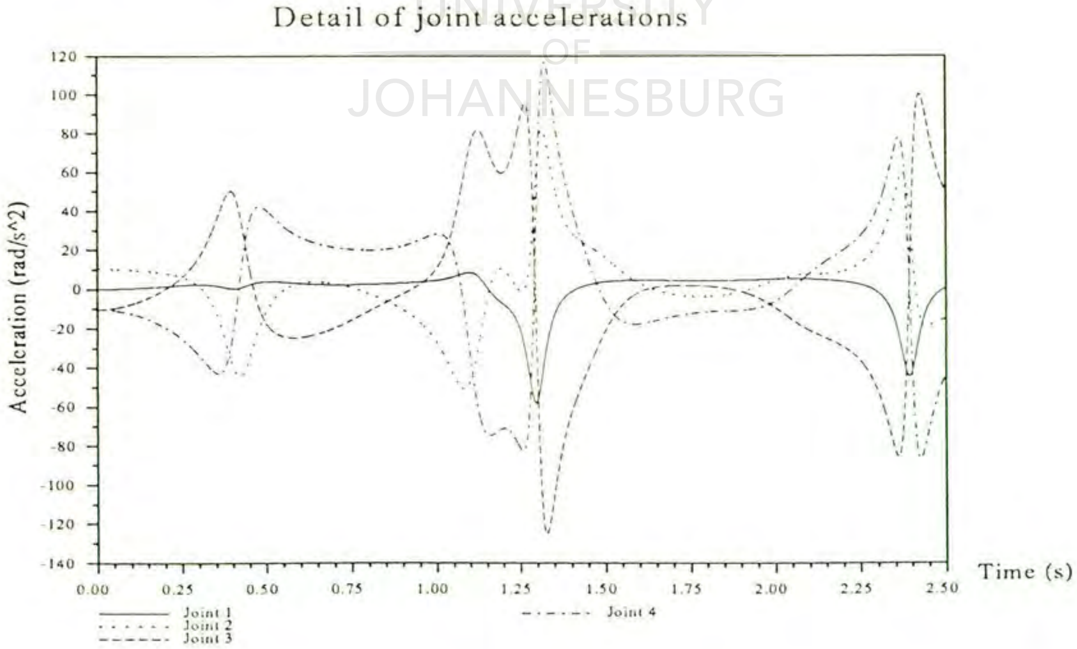


Figure 5.7: Detail of the joint accelerations.

force is added to the correctly calculated tip force to bring the tip of the manipulator back to the position it is constrained to. The position and rate feedbacks used were 60 N/m and 40 Ns/m. These values are quite small, but still relatively good results were obtained despite the simple integration techniques used. These "corrective" forces are plotted in Figure 5.8. The discrepancy between the calculated position of the tip and the position[9] it is constrained to, is plotted in Figure 5.9.

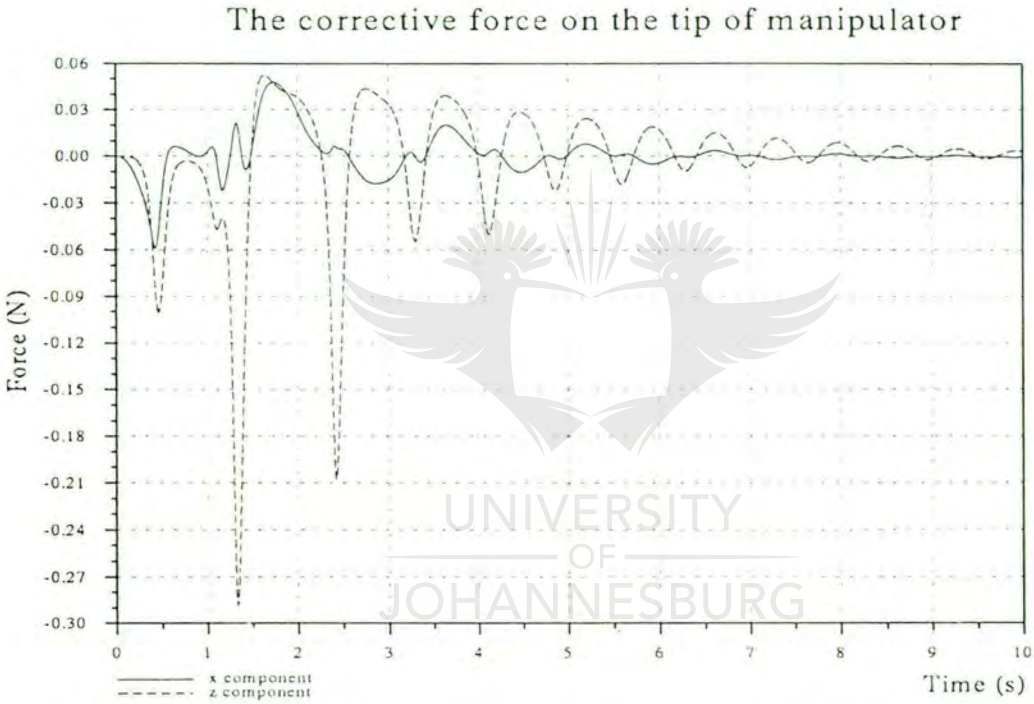


Figure 5.8: The corrective force applied to the tip of manipulator to counteract integration error.

The components of the constraint force that the tip exerts on the environment are plotted in Figure 5.10. Although the total mass of the system is only seven kilograms, peak constraint forces in the order of 230 N occurs. This is due to the high acceleration peaks, which in turn is due to the complex interaction of the links of the uncontrolled system. Detail of the tip forces during the first two and a half seconds of simulation is given in Figure 5.11.

[9] It is not the position of the end effector that is constrained, but it acceleration.

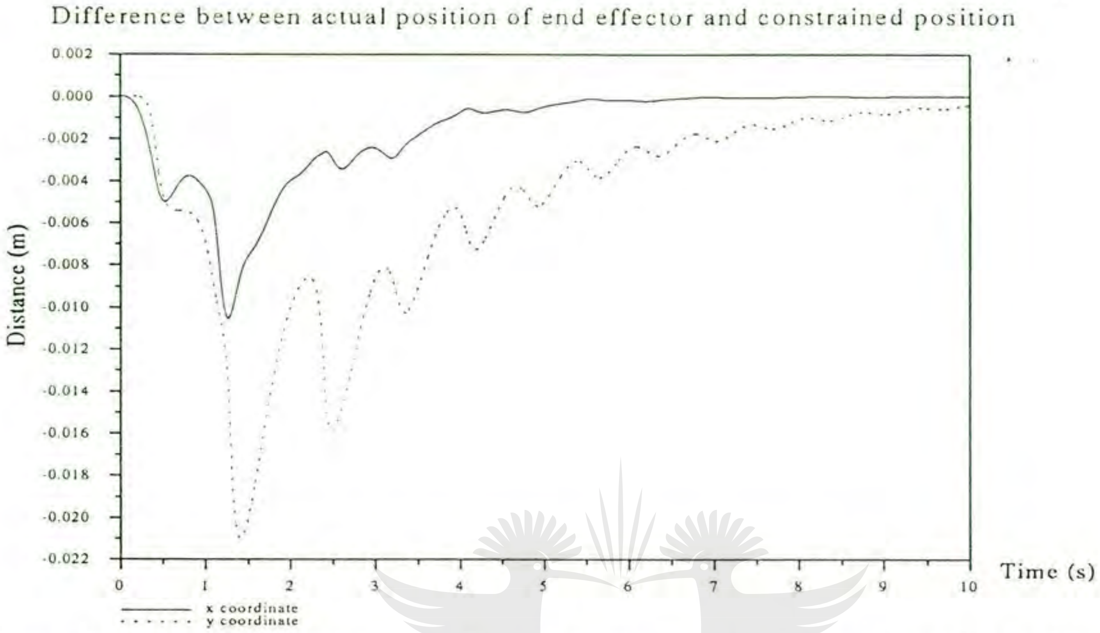


Figure 5.9: The difference between the calculated position of the tip and the position it is constrained to.

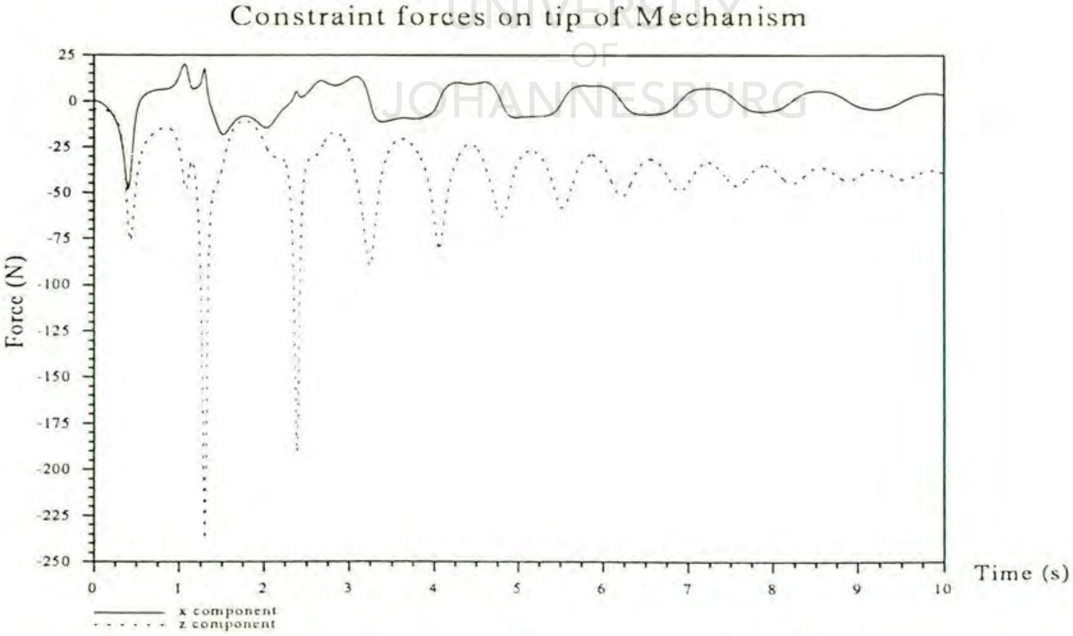


Figure 5.10: The constraint components of the tip force of the four link manipulator.

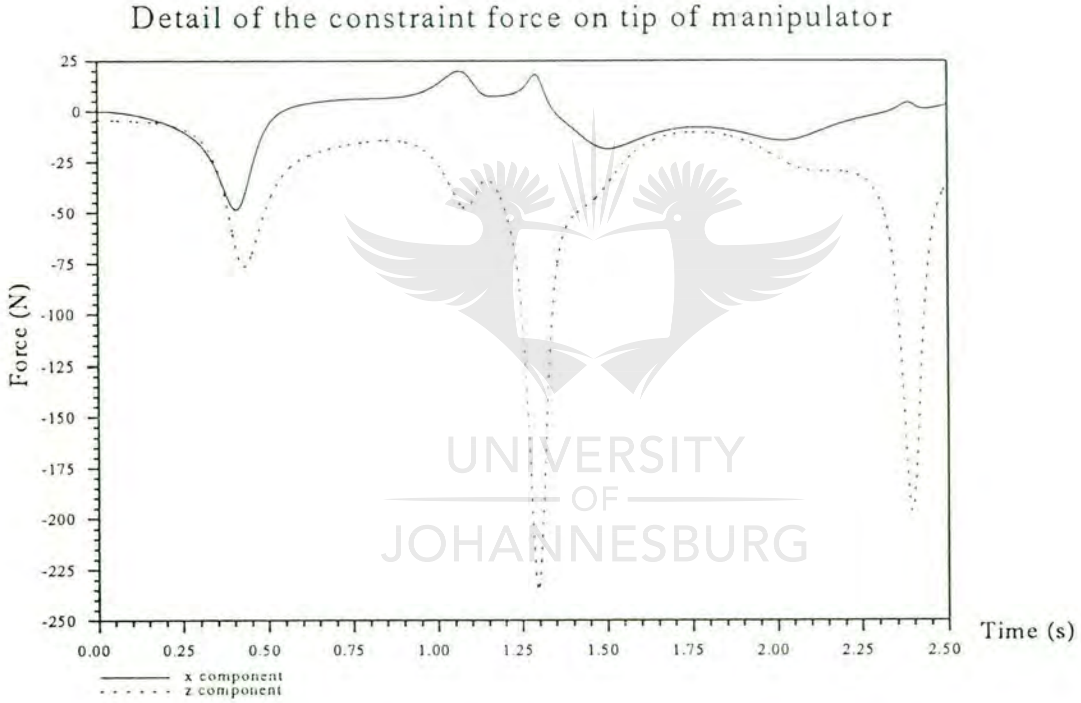


Figure 5.11: Detail of the constraint components of the tip force.

By comparing Figures 5.8 and 5.10 it is clear that the corrective force is very small compared to the actual tip force.

5.6 Summary

In this chapter it has been shown how the forward dynamic problem for single serial open or closed chains can be solved, using the spatial notation and the Newton-Euler formalism. An analytical method for the calculation of the time derivative of the Jacobian has been proposed. Implementation details like minimizing the effects of numerical integration error has also been addressed. Finally the solution for a simple four link mechanism has been worked through for the first time step of a simulation, and the results for ten simulated seconds given and very briefly discussed.

Chapter 6

Simulation of simple closed chain mechanisms

> *But if one looks for the motion of several bodies which react upon each other.... then the question is of a higher order, and the preceding principles (i.e. those known before 1742) are insufficient to solve it.*
>
> Lagrange
> 1788

6.1 Definition of simple closed-chain mechanism

Simple closed-chain mechanisms is a subset of multiple chain robotic mechanisms. Such mechanisms are frequently encountered in physical robotic systems. Examples of such systems are:

- More than one manipulator handling a common workload.
- Legged walking machines when more than one of the feet are in contact with a supporting surface.
- The fingers of a dextrous hand manipulating an object.
- A single manipulator with more than one link supporting the end-effector (high stiffness, low mass manipulators).

Many non-robotic rigid body systems can also be described as simple closed-chain mechanisms, for example the skeletal structure of humans. This can be very usefully applied to determine what forces acts on the joints of a human when performing certain tasks. One can well imagine that this can be applied in sport research, to analyze and improve the performance in field events, and to analyze possible new gymnastics sequences. What all these systems have in common is a structure characterized by a group of actuated chains all supporting a common body, called the reference member.

As can be expected from the fact that the movement of all the chains are coupled by the reference member, the dynamic simulation of such systems is even more difficult and computationally intensive than that of single closed chains. To solve the behaviour of the system, one has to determine the acceleration of the reference member, the forces that the chains exert on the reference member, and the acceleration of the individual chains. Fortunately, many concepts used to solve the single chain simulation problem can be extended to simple closed-chain mechanisms.

Many schemes to solve this problem have been proposed. Many of the first methods proposed were difficult to apply or computationally inefficient. The method described by Oh and Orin [26] is applicable to simple closed-chain mechanisms with m chains of N links each. A large system of linear algebraic equations is formed by combining the equations of motion for each chain, the motion equation for the reference body, and the kinematic constraint equations for the interactions between the chains and the reference member. The approach is straightforward, but the size of the linear system that has to be solved gives this approach a complexity of $O(m^3N^3)$.

Brandl, Johanni and Otter [27] developed an algorithm for the simulation of multi-body systems with kinematic loops. This algorithm makes use of a general joint model and is quite general. Its computational complexity is linear in the total number of bodies in the system.

A connection between filtering and smoothing theory and rigid body dynamics has been established [28]. The use of linear operator methods [29, 30] has made further progress in reducing the computational load, and is a general and powerful mathematical tool. However, the intuitive physical interpretation of these equations is difficult because of the algebraic manipulation techniques used.

6.2 A model for simple closed-chain mechanisms

The structure of simple closed-chain mechanisms consists of m actuated chains which support a single common reference member [26]. The supporting chains are serial-link chains and therefore free of internal closed loops. If the reference member is removed, all closed loops in the system are broken.

The $k-$th chain ($k = 1, 2, ..., m$) may have an arbitrary number of links, general joints, and N_k degrees of freedom (where the number of degrees of freedom of the interaction between the $k-$th chain and the reference member is excluded from N_k). The number of degrees-of-freedom of the $k-$th chain can then be less than, equal to, or greater than six. Every chain in the system can have a different number of links and different joints.

The reference member is called member r. The interaction between the $k-$th chain and the reference member is also described by the general joint model described earlier. Thus both powered and unpowered contacts can exist at the interactions. The part of the $k-$th chain that interacts with reference member is called the tip of the $k-$th chain. The opposite terminal link of each chain can interact with a support surface. This link is called link number one of chain k. The general joint model is also used to describe the interaction with the support surface. This joint is called joint number one on the of chain k. The links and joints are thus numbered from the support surface towards the reference member. The link that interacts with the reference member is called link N, and its far end is thus the tip of chain k. The interaction with the reference member is not included as a joint of the $k-$th chain. The motion of the support surface is assumed to be known. In many cases it is actually prescribed.

Depending on the nature of the interactions between the chains and the reference member and between the chains and the support surface(s), simple closed-chain mechanisms can be divided into two basic types, called Type 0 and Type 1 respectively [26]. A Type 0 mechanism is one where the first link of each chain is connected to the support surface by an actuated joint, while the tip of each chain interacts with the reference member through an unpowered contact. Multiple manipulators manipulating a common load and dextrous hands are examples of such mechanisms. On the other hand, in a Type 1 mechanism the first link of each chain interacts through an unpowered contact with the support surface and the last link is connected to the reference member by an actuated joint. Multi-legged vehicles can be modeled as Type 1 mechanisms. Both Type 0 and Type 1 simple closed-chain mechanisms are modeled in the same way, and both can be simulated by the dynamics algorithms described later in this chapter.

Details such as the location and orientation of coordinate frames can be application dependent, and attention to this will be given in a later section of this chapter. For now, the general problem is studied at a higher analytical level, which leads to better physical insight.

6.3 System dynamic equations

6.3.1 Equations of motion for each chain

Even though each chain is connected to the reference member, the dynamic equation for each chain is the same as for a single closed chain. The force the tip of each chain exerts on the reference member completely accounts for the influence the reference member has on all the chains. These forces are not known at this stage, but the dynamic equation for the $k-$th chain, in the joint space, can still be written as:

$$\Gamma_k = \mathbf{H}_k\ddot{\mathbf{q}}_k + \mathbf{C}_k\dot{\mathbf{q}}_k + \mathbf{G}_k + \mathbf{J}_k^T\mathbf{f}_k \tag{6.1}$$

All the symbols have been defined in chapter 5. Remember that $\mathbf{J}_k$ is the Jacobian matrix for the $k-$th chain, evaluated at its tip, and $\mathbf{f}_k$ is the spatial force (6×1) that the tip of the $k-$th chain exerts of the reference member. For the direct dynamics problem the unknowns in equation 6.1 are the joint accelerations, $\ddot{\mathbf{q}}_\mathbf{k}$, and the components of the force, $\mathbf{f}_k$, that lies in the constrained directions of the interaction with the reference member. The other quantities on the right hand side of equation 6.1 are only functions of the general joint positions and velocities, $\ddot{\mathbf{q}}_\mathbf{k}$ and $\dot{\mathbf{q}}_\mathbf{k}$.

As shown in chapter 5, both the joint acceleration and the spatial tip acceleration vectors of each chain can be partitioned into the open chain solution and a function of the tip force:

$$\begin{aligned}
\ddot{\mathbf{q}}_k &= (\ddot{\mathbf{q}}_k)_{\text{open}} - \mathbf{\Omega}_k\mathbf{f}_k && (6.2)\\
\ddot{\mathbf{x}}_k &= (\ddot{\mathbf{x}}_k)_{\text{open}} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k && (6.3)\\
& && (6.4)
\end{aligned}$$

$\mathbf{\Lambda}_k^{-1}$ is the inverse operational space inertia matrix for chain k, which is always uniquely defined for each chain, although it is a function of the joint positions of the particular chain. As discussed in chapter 5, the open-chain accelerations can be uniquely determined from the present state and inertial properties of the chain, the applied joint forces and the motion of the support surface (base). In the simulation algorithm for simple closed-chain mechanisms, these accelerations can be solved using any appropriate open-chain direct dynamics algorithm, one of which can be found in chapter 5. $\mathbf{\Omega}_k$ and $\mathbf{\Lambda}_k^{-1}$ can be determined because the general joint positions are known. The only unknown that remains to be solved before the motion of each of the chains in the simple closed-chain mechanism can be solved for, are the tip forces $\mathbf{f}_k$. Intuitively one expects that these forces can be determined by considering the inertial properties of the reference member and all the chains. If this is combined with the kinematic constraints between the reference member and the tips of the chains, and the external forces of the reference member, these forces can indeed be solved for. As the first step towards this, the dynamic equation for the motion of the reference member has to be considered.

6.3.2 Dynamic equation for the reference member

The dynamic behaviour of the reference member will also be described using spatial quantities. This is done by constructing a force balance for the reference body. The resultant force acting on that member is the sum of the tip forces exerted by the individual chains on it and any external forces (like gravitational) acting on it. Since spatial force includes both the linear forces and moments, the resultant force can be written very compactly as:

$$\mathbf{F}_r = \sum_{k=1}^{m} {}^r\mathbf{f}_k + \mathbf{g}_r \tag{6.5}$$

where

$\mathbf{F}_r$ = 6×1 resultant spatial force on the reference member.

${}^r\mathbf{f}_k$ = 6×1 spatial force applied by chain k to the reference member.

$\mathbf{g}_r$ = 6×1 external spatial force acting on the reference member, which must include gravity, where present

and all the spatial forces in equation 6.5 is measured with respect to the coordinate frame fixed to the reference member (frame r). The relationship between the resultant force on the reference member and its acceleration is given by the Newton-Euler dynamic equations, which reduces to the following single equation when spatial vectors are used:

$$\mathbf{F}_r = \mathbf{I}_r\mathbf{a}_r + \mathbf{b}_r, \tag{6.6}$$

where

$\mathbf{b}_r$ = $\begin{bmatrix} \omega_r \times \bar{\mathbf{I}}_r\omega_r \\ m_r\omega_r \times (\omega_r \times \mathbf{s}_r) \end{bmatrix}$.

$\mathbf{I}_r$ = 6×6 spatial inertia of the reference member.

$\mathbf{a}_r$ = 6×1 spatial acceleration of the reference member.

ω_r = 3×1 angular velocity vector of the reference member.

$\bar{\mathbf{I}}_r$ = 3×3 inertia tensor of the reference member.

m_r = mass of the reference member.

$\mathbf{s}_r$ = 3×1 position vector of the center of mass of the reference member relative to the origin of frame r.

and all the quantities above (excluding the mass, which is frame independent) are expressed in coordinate frame r. Now equations 6.5 and 6.6 can be combined to yield the following dynamic equation for the reference member:

$$\sum_{k=1}^{m} {}^{r}\mathbf{f}_k + \mathbf{g}_r = \mathbf{I}_r \mathbf{a}_r + \mathbf{b}_r \tag{6.7}$$

The unknown variables in equation 6.7 are the acceleration of the reference member, $\mathbf{a}_r$, and the components of the tip forces, ${}^{r}\mathbf{f}_k$, which lies in the constrained directions of the interactions of the chains with the reference member. The above equation is obviously not sufficient to solve for the acceleration of the reference member. However, when this equation is combined with the partitioned closed-chain operational space dynamic equation, equation 6.4, the whole system can be solved. This is what will be illustrated in the following subsection.

6.3.3 Simulation algorithm for system

The derivation of a simulation algorithm for simple closed-chain mechanisms presented by Lilly [2], is briefly summarized below. The system modeling has already been described. The first step in the derivation is to derive an explicit relationship between the spatial acceleration of the tip of chain k and the spatial acceleration of the reference member.

The spatial acceleration of the interaction point of chain k with the reference member, which is also the spatial acceleration of the far side (as seen by the chain) of the general joint between the chain and the reference member, is the sum of the acceleration of the tip of chain k and the relative acceleration between the tip and the reference member:

$$\mathbf{a}_{r,k} = \ddot{\mathbf{x}}_k + \ddot{\mathbf{x}}_k^{\text{relative}}, \tag{6.8}$$

where $\ddot{\mathbf{x}}_k^{\text{relative}}$ is the relative acceleration between the tip and reference body. Constraints are placed on this relative acceleration by the joint between the tip and reference member. Thus the relative acceleration is resolved in the constrained and free spaces of the joint:

$$\ddot{\mathbf{x}}_k^{\text{relative}} = \phi_k \alpha_k + \phi_k^c \alpha_k^c, \tag{6.9}$$

where ϕ_k and ϕ_k^c are the motion and constraint space of the general joint representing the interaction between the tip of chain k and the reference member. Similarly, α_k and α_k^c are the corresponding components of the relative acceleration. As explained in section 5.2.4, for a tip contact ϕ_k, ϕ_k^c and α_k^c are known. Often α_k^c is zero. Equation 6.9 can be substituted into equation 6.8 to yield:

$$\mathbf{a}_{r,k} = \ddot{\mathbf{x}}_k + \phi_k \alpha_k + \phi_k^c \alpha_k^c \tag{6.10}$$

Since the reference member is treated as a rigid body, the spatial acceleration of the interaction point of chain k on the reference body can be obtained by transforming the spatial acceleration of the coordinate origin of the reference body to the interaction point, and adding a bias vector to account for centripetal and Coriolis effects. In view of the above, the acceleration of interaction point k can also be written as:

$$\mathbf{a}_{r,k} = {}^k\mathbf{X}_r \mathbf{a}_r + \zeta_{r,k}, \tag{6.11}$$

where ${}^k\mathbf{X}_r$ is the spatial transformation from coordinate frame r to the coordinate frame associated with the general joint at the tip of chain k. The 6×1 bias vector $\zeta_{r,k}$ is a function of the velocity of the reference member and the position of the interaction point (see section 6.4.3):

$$\zeta_{r,k} = {}^k\mathbf{X}_r \begin{bmatrix} 0_{3\times1} \\ \omega_r \times (\omega_r \times \mathbf{p}_k) \end{bmatrix}. \tag{6.12}$$

By combining equations 6.10 and 6.11 an explicit relationship between $\ddot{\mathbf{x}}_k$ and $\mathbf{a}_r$ is established:

$$\ddot{\mathbf{x}}_k + \phi_k \alpha_k + \phi_k^c \alpha_k^c = {}^k\mathbf{X}_r \mathbf{a}_r + \zeta_{r,k} \tag{6.13}$$

The importance of equation 6.13 is that it matches the spatial accelerations at the interaction point. The only unknowns are $\ddot{\mathbf{x}}_k$ and $\mathbf{a}_r$, the other quantities are known from the definition of the simulation problem and the present state of the reference member.

The next step in solving the simple closed-chain problem is to find an explicit relationship between the spatial force exerted by the tip of chain k on the reference member, $\mathbf{f}_k$, and the spatial acceleration of the reference member. In equation 6.4 we already have the relationship between the tip force and the spatial acceleration of the tip, and equation 6.13 relates the spatial acceleration of the tip to the acceleration of the reference member. We are almost there. The only problem is that equation 6.13 also has the unconstrained joint acceleration as an unknown. In the same way as with single closed chains, this acceleration can be eliminated by projecting equation 6.13 on the constraint space of the tip:

$$(\phi_k^c)^T \left[\ddot{\mathbf{x}}_k + \phi_k \alpha_k + \phi_k^c \alpha_k^c\right] = (\phi_k^c)^T \left[{}^k\mathbf{X}_r \mathbf{a}_r + \zeta_{r,k}\right]. \tag{6.14}$$

Because the motion and constraint spaces are by definition orthogonal for the general joint, $(\phi_k^c)^T \phi_k = \mathbf{0}$ and $(\phi_k^c)^T \phi_k^c = \mathbf{1}$. The above equation thus reduces to:

$$(\phi_k^c)^T \ddot{\mathbf{x}}_k + \alpha_k^c = (\phi_k^c)^T \left[{}^k\mathbf{X}_r \mathbf{a}_r + \zeta_{r,k}\right] \tag{6.15}$$

Now we are ready to substitute equation 6.4 in order to establish the relationship between the tip-force and reference member acceleration:

$$
\begin{aligned}
(\phi_k^c)^T \left[(\ddot{\mathrm{x}}_k)_{\mathrm{open}} - \mathbf{\Lambda}_k^{-1} \mathrm{f}_k \right] &= (\phi_k^c)^T \left[{}^k\mathbf{X}_r \mathrm{a}_r + \zeta_{r,k} \right] - \alpha_k^c & (6.16) \\
\left[(\phi_k^c)^T \mathbf{\Lambda}_k^{-1} \right] \mathrm{f}_k &= \left[\alpha_k^c - (\phi_k^c)^T \zeta_{r,k} + (\phi_k^c)^T (\ddot{\mathrm{x}}_k)_{\mathrm{open}} \right] \\
&\quad - \left[(\phi_k^c)^{T\,k} \mathbf{X}_r \right] \mathrm{a}_r & (6.17)
\end{aligned}
$$

The components of f_k that lies in the free space of the interaction are given for the simulation problem. We are thus interested in finding the relationship between the unknown force components that lie in the constrained directions of the general joint at the tip. In the same way as the relative acceleration, the tip-force spatial vector can also be resolved in the motion and free spaces of the general joint at the tip of chain k:

$$
\mathrm{f}_k = \phi_k \mathrm{h}_k + \phi_k^c \mathrm{h}_k^c, \qquad (6.18)
$$

where h_k is the vector of known force components in the free directions, and h_k^c is the vector of unknown force components in the constraint directions. The above equation and equation 6.17 can be combined and an explicit relationship between the unknown components of the tip-force and the spatial acceleration of the reference member results:

$$
\begin{aligned}
(\phi_k^c)^T \mathbf{\Lambda}_k^{-1} \left[\phi_k \mathrm{h}_k + \phi_k^c \mathrm{h}_k^c \right] &= \left[\alpha_k^c - (\phi_k^c)^T \zeta_{r,k} + (\phi_k^c)^T (\ddot{\mathrm{x}}_k)_{\mathrm{open}} \right] \\
&\quad - \left[(\phi_k^c)^{T\,k} \mathbf{X}_r \right] \mathrm{a}_r & (6.19) \\
\left[(\phi_k^c)^T \mathbf{\Lambda}_k^{-1} \phi_k^c \right] \mathrm{h}_k^c &= \left\{ \alpha_k^c - (\phi_k^c)^T \left[\zeta_{r,k} + (\ddot{\mathrm{x}}_k)_{\mathrm{open}} + \mathbf{\Lambda}_k^{-1} \phi_k \mathrm{h}_k \right] \right\} \\
&\quad - \left[(\phi_k^c)^{T\,k} \mathbf{X}_r \right] \mathrm{a}_r & (6.20) \\
\mathrm{M}_k^{-1} \mathrm{h}_k^c &= \mathbf{S}_k - \left[(\phi_k^c)^{T\,k} \mathbf{X}_r \right] \mathrm{a}_r, & (6.21)
\end{aligned}
$$

where $\mathbf{M}_k$ and $\mathbf{S}_k$ are known.

If $n_{c,k}$ is the number of degrees of constraint for the general joint at the tip of chain k, then M_k^{-1} is the $n_{c,k} \times n_{c,k}$ coefficient matrix for h_k^c in the linear system of equation 6.21. M_k^{-1} may be singular, in which case equation 6.21 does not have a unique solution. This happens when the base for the column space of ϕ_k^c (which is simply the columns of ϕ_k^c) is not contained in the base for the column space of $\mathbf{\Lambda}_k^{-1}$. The implications of this and the solution of the system in such cases will be discussed in the next section on numerical implementation. Physically, $\mathbf{M}_k$ is the part of the inertia of chain k, as seen at its tip, that is reflected across the general joint at the tip to the reference member. The pre- and post multiplication of the inverse operational space inertia matrix by ϕ_k^c accomplishes this as only the components of the operational space inertia matrix that corresponds with the constrained directions of the general joint can be reflected across the joint. If the tip of chain k is rigidly attached to the reference member, ϕ_k^c is the 6×6 identity matrix, and $\mathbf{M}_k^{-1}$ will be exactly equal to the inverse operational space inertia matrix for chain k, $\mathbf{\Lambda}_k^{-1}$.

Even though $\mathbf{M}_k^{-1}$ may be singular, a solution to equation 6.21 can be found. A $\mathbf{h}_k^c$ (it may not be unique) can thus be written as a function of $\mathbf{a}_r$. We will represent this fact on an analytical level by rewriting equation 6.21 as:

$$\mathbf{h}_k^c = \mathbf{M}_k\mathbf{S}_k - \left[\mathbf{M}_k(\phi_k^c)^T\,{}^k\mathbf{X}_r\right]\mathbf{a}_r, \tag{6.22}$$

even though $\mathbf{M}_k$ may not be the inverse of $\mathbf{M}_k^{-1}$. Given the general solution for $\mathbf{h}_k^c$ in the above equation, one can write the tip-force vector $\mathbf{f}_k$ as:

$$\begin{aligned} \mathbf{f}_k &= \phi_k\mathbf{h}_k + \phi_k^c\mathbf{h}_k^c, & (6.23)\\ &= \left[\phi_k + \phi_k^c\mathbf{M}_k\mathbf{S}_k\right] - \left[\phi_k^c\mathbf{M}_k(\phi_k^c)^T\,{}^k\mathbf{X}_r\right]\mathbf{a}_r, & (6.24)\\ &= \mathbf{P}_k - \mathbf{R}_k\mathbf{a}_r. & (6.25) \end{aligned}$$

$\mathbf{P}_k$ and $\mathbf{R}_k$ are of dimension 6×1 and 6×6 respectively and both can be calculated from the foregoing.

The third step in solving the simple closed-chain problem can now be taken. In equation 6.25 we have related the tip-force of chain k, $\mathbf{f}_k$, to the acceleration of the reference member, $\mathbf{a}_r$. This can now be used with the dynamic equation (equation 6.7), to eliminate the tip force, and the resulting equation has only $\mathbf{a}_r$ as unknown and the system can be solved. First, rewrite equation 6.7 as:

$$\sum_{k=1}^{m}({}^k\mathbf{X}_r)^T\mathbf{f}_k + \mathbf{g}_r = \mathbf{I}_r\mathbf{a}_r + \mathbf{b}_r, \tag{6.26}$$

where

$$\begin{aligned} {}^r\mathbf{f}_k &= {}^r\mathbf{X}_k.\mathbf{f}_k & (6.27)\\ &= ({}^k\mathbf{X}_r)^T\mathbf{f}_k, & (6.28) \end{aligned}$$

has been used. Then equation 6.25 can be substituted into equation 6.26 to obtain:

$$\sum_{k=1}^{m}({}^k\mathbf{X}_r)^T(\mathbf{P}_k - \mathbf{R}_k\mathbf{a}_r) = \mathbf{I}_r\mathbf{a}_r + \mathbf{b}_r - \mathbf{g}_r. \tag{6.29}$$

Collecting terms with $\mathbf{a}_r$ yields:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m}({}^k\mathbf{X}_r)^T\mathbf{R}_k\right]\mathbf{a}_r = \left[\sum_{k=1}^{m}({}^k\mathbf{X}_r)^T\mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right]. \tag{6.30}$$

Equation 6.30 has only the spatial acceleration of the reference member, $\mathbf{a}_r$, as unknown. One can find a solution from the set of linear algebraic equations. An explicit analytical solution for $\mathbf{a}_r$ can be written:

$$\mathbf{a}_r = \left[\mathbf{I}_r + \sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{R}_k\right]^{-1}\left[\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right] \tag{6.31}$$

The inversion (which is always 6×6) can always be calculated, but if $\mathbf{M}_k^{-1}$ does not have a rank of six, the results would be meaningless. This and the solution procedure is discussed in the next section.

By expanding the above equation, one can obtain more physical insight:

$$\mathbf{a}_r = \left[\mathbf{I}_r + \sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\phi_k^c\mathbf{M}_k(\phi_k^c)^T\,{}^{k}\mathbf{X}_r\right]^{-1}\left[\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right]. \tag{6.32}$$

The first term on the right of the above equation represents the combined inertial properties of the reference member and all the chains. The inertial properties of the chains that are reflected across the joints is transformed to the reference body frame before being added to the inertia of the reference member. This term thus represents the effective operational space inertia of the simple closed chain, defined at the origin of frame r. The second term is the sum of all the spatial forces acting on the reference member.

After the spatial acceleration of the reference member, $\mathbf{a}_r$ has been determined, the spatial forces exerted by the tips if the chains on the reference member can be calculated from equation 6.25, which is repeated here for convenience:

$$\mathbf{f}_k = \mathbf{P}_k - \mathbf{R}_k\mathbf{a}_r. \tag{6.33}$$

The fact that $\mathbf{f}_k$ is known allows us to treat chain k as a single closed chain with a known tip force. The single closed-chain joint acceleration has been partitioned into terms containing the open chain acceleration and tip force, and the closed-chain joint accelerations for each chain k can be determined from equation 6.4, also repeated for convenience:

$$\ddot{\mathbf{q}}_k = (\ddot{\mathbf{x}}_k)_{\text{open}} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k. \tag{6.34}$$

Having computed the spatial acceleration of the reference member, the tip forces of each chain, and the joint accelerations for each chain, the system has been solved and the next state of the system can be computed by integration. For more detail on this, see the next section. The simulation algorithm can be summarized as:

1. Calculate the *open-chain joint accelerations* and *operational space inertias* for each chain.

2. Calculate the *spatial acceleration of the reference member.*
3. Calculate the *spatial tip forces* exerted by the chains on the reference member.
4. Calculate the *closed-chain joint accelerations* for each chain.
5. *Integrate* to the next state.

6.4 Details of numerical implementation

Many important details have been ignored in the previous section, which can make the implementation of the simulation algorithm difficult. A few of them are addressed below:

6.4.1 The transformation between the reference member frame and tip frames

The coordinate frame associated with the general joint at the tip of chain k that represents the interaction of the chain with the reference body can either be fixed to the end of the last link in the chain, or on the far side of the joint (nearest to the reference body). In order to maintain consistency with the single-closed chain algorithm, let us choose to fix it to the tip of chain k. For the simulation algorithm, we need to know the spatial transformation from the reference body frame to the frame fixed to the tip of each chain k.

First let us consider the 3×3 rotation transformation. The base frame for the system is called frame 0. The rotation transformation from the base frame to the tip frame of chain k can be written as:

$$({}^{\text{tip}}A_0)_k = ({}^{\text{tip}}A_N)_k \dots ({}^2A_1)_k({}^1A_0)_k, \tag{6.35}$$

by making all the transformations between the coordinate frames of chain k. This transformation can also be written by making the transformations to the reference member and then to the tip of chain k:

$$({}^{\text{tip}}A_0)_k = {}^kA_r\,{}^rA_0. \tag{6.36}$$

From the above equation, the transformation from the reference frame to the tip frame of chain k can be calculated from other rotations which have to be known or calculated for the simulation in any case:

$$\begin{aligned} {}^kA_r &= ({}^{\text{tip}}A_0)_k({}^rA_0)^{-1} && (6.37)\\ &= ({}^{\text{tip}}A_0)_k({}^rA_0)^T && (6.38)\\ &= \left[({}^{\text{tip}}A_N)_k \dots ({}^2A_1)_k({}^1A_0)_k\right]({}^rA_0)^T && (6.39) \end{aligned}$$

The interaction position of the tip of chain k with the reference member, is described by the position vector, $\mathbf{p}_k$, of this point relative to the origin of frame r, and is expressed in the reference body frame. This position vector is always constant. Since the rotation transformation from frame r to the tip frame of chain k, and the position of the origin of the tip frame is known, the spatial transformation is simply:

$$ {}^k\mathbf{X}_r = \begin{bmatrix} {}^kA_r & 0 \\ {}^kA_r\tilde{\mathbf{p}}_k^T & {}^kA_r \end{bmatrix}. \qquad (6.40)$$

6.4.2 The position of the reference body

The orientation of the reference body and the position of a point on it is known at the start of the simulation. At each time step in the simulation, the present state of the system is known. The spatial velocity of the reference member is thus also known. The spatial acceleration of the reference member is also calculated in this time step, and from this the next state spatial velocity can be found. The first three components of the spatial velocity is the angular velocity:

$$\mathbf{v}_r = \begin{bmatrix} \omega_r \\ v_r \end{bmatrix}. \qquad (6.41)$$

From the angular velocity of the reference member, an instantaneous rotation axis for it can be found:

$$\hat{k} = \frac{\omega_r}{\|\omega_r\|}, \qquad (6.42)$$

and a rotation speed can also be found:

$$\dot{\theta}_r = \|\omega_r\|. \qquad (6.43)$$

In a similar way an instantaneous direction of linear motion and speed can be found from the linear velocity (see Figure 6.1). In the simulation problem, the movement of the reference member in each time step can be approximated as rotation around some instantaneous axis by an angle, and movement in an instantaneous direction by some distance. The rotation axis can be approximated as the mean of the present state axis and the next state axis:

$$\dot{\hat{k}} = \frac{\hat{k}_{t+\delta t} - \hat{k}_t}{\delta t} \qquad (6.44)$$

The rotation angle can be approximated as:

$$\theta_r = (\dot{\theta}_{r,t+\delta t} + \dot{\theta}_{r,t})\frac{\delta t}{2} \qquad (6.45)$$

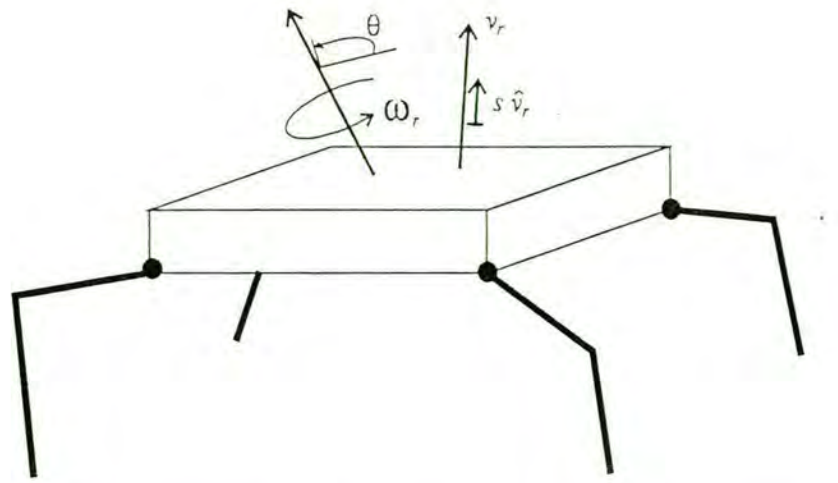


Figure 6.1: Description of the motion of the reference member.

Exactly the same can be done for the linear velocity. To arrive at more compact expressions, let us define the spatial velocity direction as:

$$\begin{aligned} \hat{\mathrm{k}} &= \begin{bmatrix} \frac{\omega_r}{\|\omega_r\|} \\ \frac{v_r}{\|v_r\|} \end{bmatrix} && (6.46) \\ &= \begin{bmatrix} \hat{\omega}_r \\ \hat{v}_r \end{bmatrix}. && (6.47) \end{aligned}$$

The rotational and linear speeds are also combined as:

$$\dot{d} = \begin{bmatrix} \|\omega_r\| \\ \|v_r\| \end{bmatrix}. \qquad (6.48)$$

With the above definitions, the approximation for the direction of the spatial velocity for simulation purposes is given by:

$$\bar{\hat{\mathrm{k}}} = \frac{\hat{\mathrm{k}}_{t+\delta t} - \hat{\mathrm{k}}_t}{\delta t}. \qquad (6.49)$$

and the approximation for the distance traveled by the reference member in the simulation time step, is given by:

$$\begin{aligned} d &= (\dot{d}_{t+\delta t} + \dot{d}_t)\frac{\delta t}{2} && (6.50) \\ &= \begin{bmatrix} \theta \\ s \end{bmatrix} && (6.51) \end{aligned}$$

The change of orientation of the reference member is represented by the 3 × 3 rotation matrix ${}^{t+\delta t}A_t$. This is approximated as a general rotation around a fixed axis. The first three components of $\bar{\hat{\mathrm{k}}}$ (the components of $\hat{\omega}_r$) is called o_x, o_y and o_z. The rotation is then given by:

$$
{}^{t+\delta t}A_t = \begin{bmatrix} o_x o_x v\theta + c\theta & o_y o_x v\theta - o_z s\theta & o_z o_x v\theta + o_y s\theta \\ o_x o_y v\theta + o_z s\theta & o_y o_y v\theta + c\theta & o_z o_y v\theta - o_x s\theta \\ o_x o_z v\theta - o_y s\theta & o_y o_z v\theta + o_x s\theta & o_z o_z v\theta + c\theta \end{bmatrix}, \qquad (6.52)
$$

where $v\theta = \operatorname{vers}\theta = (1 - \cos\theta)$, $s\theta = \sin\theta$ and $c\theta = \cos\theta$. Given the rotation transformation from the base frame to the reference frame at a time step t, the rotation transformation from the base frame to the reference body frame at the next time step can be obtained:

$$
{}^{r}A_0\big|_{t+\delta t} = {}^{t+\delta t}A_t \, {}^{r}A_0\big|_t \,. \qquad (6.53)
$$

The above rotation transformation completely describes the orientation of the reference member in the next time step of the simulation. The linear position of the reference member is described by the position of the origin of frame r, which is given by:

$$
\mathrm{p}_r\big|_{t+\delta t} = \mathrm{p}_r\big|_t + s\hat{v}_r. \qquad (6.54)
$$

6.4.3 Calculation of the bias acceleration vectors $\zeta_{r,k}$

The bias spatial acceleration vector, $\zeta_{r,k}$, is expressed in the tip frame of chain k. However, it is easier to calculate it in frame r and then transform this. The centripetal acceleration of the interaction point of chain k with the reference member, expressed in frame r, is simply:

$$
a_{\text{centripetal}} = \omega_r \times (\omega_r \times \mathrm{p}_k) \qquad (6.55)
$$

The position of the interaction point is usually fixed relative to frame r, and then has no velocity relative to frame r. Therefore, the interaction point has no Coriolis acceleration relative to frame r. The bias acceleration vector is then simply:

$$
\zeta_{r,k} = {}^{k}\mathbf{X}_r \begin{bmatrix} 0_{3\times 1} \\ \omega_r \times (\omega_r \times \mathrm{p}_k) \end{bmatrix}. \qquad (6.56)
$$

6.4.4 Singularities in M_k^{-1}

As stated earlier, M_k^{-1} is singular if the base for ϕ_k^c is not contained in the base for Λ_k^{-1}. This follows from the definition of M_k^{-1}:

$$
\mathrm{M}_k^{-1} = (\phi_k^c)^T \Lambda_k^{-1} (\phi_k^c) \qquad (6.57)
$$

Recall that $n_{c,k}$ is the number of constraint directions of the general joint at the tip of chain k, and that the dimension of $\mathbf{M}_k^{-1}$ is $n_{c,k} \times n_{c,k}$. Let us further clarify when $\mathbf{M}_k^{-1}$ is singular. If Λ_k^{-1} has full column rank, namely six, the base for ϕ_k^c will always be contained in the column space of Λ_k^{-1}, and $\mathbf{M}_k^{-1}$ can not be singular. A necessary, but not sufficient condition for $\mathbf{M}_k^{-1}$ being singular, is thus that Λ_k^{-1} must have a rank less than six.

The physical interpretation of Λ_k^{-1} having a rank less than six, is simply that the end effector cannot be moved in all directions, and/or rotated in all directions. In certain directions, the magnitude of a force applied to the tip of the chain has no effect on the motion of the chain. This can be seen as the chain having an infinite inertia in those directions. If the constraint space of the general joint at the tip of chain k includes more than zero of these directions in which the tip of chain k cannot move, this infinite inertia in those directions is reflected across the joint to the reference member. The inertia of the k-th chain as felt by the reference member is therefore also infinite in some direction(s) – the reference member cannot move in that direction(s) – and $\mathbf{M}_k^{-1}$ does not have full column rank and must be singular. The condition stated in the first paragraph of this section is therefore sufficient for determining whether $\mathbf{M}_k^{-1}$ is singular or not.

For the simulation algorithm, we still need so be able to compute $\mathbf{M}_k\mathbf{S}_k$. If we set the reference member acceleration, $\mathbf{a}_r$, equal to zero in equation 6.21, and solve for $\mathbf{h}_k^c$, the solution would be exactly $\mathbf{M}_k\mathbf{S}_k$. If $\mathbf{M}_k^{-1}$ is singular, there would not be a unique solution to $\mathbf{h}_k^c$. The system can be solved with a linear system solver (e.g. Gauss-Jordan reduction of the augmented matrix). The solution would express the leading variables as linear functions of the free variables [15]. The general solution for $\mathbf{h}_k^c$ is then the sum of the homogeneous and a particular solution. The free variables would be the forces corresponding to the directions in which the reference member cannot move. They have no influence on the motion of the system and can thus take on any value. This is consistent with the mathematical solution of the linear system. The simplest solution is obtained by setting the homogeneous solution to zero. This is a valid solution and it is perfectly logical to set the forces that has no influence on the motion of the mechanism equal to zero. The particular solution of:

$$\mathbf{M}_k^{-1}\mathbf{h}_k^c = \mathbf{S}_k \tag{6.58}$$

can thus be used as $\mathbf{M}_k\mathbf{S}_k$ in the simulation algorithm.

Another quantity needed in the simulation algorithm, $\left[\mathbf{M}_k(\phi_k^c)^T\, {}^k\mathbf{X}_r\right]$, can also be calculated from equation 6.21, after $\mathbf{M}_k\mathbf{S}_k$ has been calculated. This can be done by solving the equation with all components of $\mathbf{a}_r$ set equal to 1, except those corresponding to the directions in which the reference body cannot move due to chain

k. If $\mathbf{b}$ is the base for the column space of $\mathbf{M}_k^{-1}$, and $rk = rank(\mathbf{M}_k^{-1})$ then $\mathbf{a}_r$ can be set equal to $\mathbf{a}_r = \mathbf{b}\mathbf{1}_{rk\times 1}$ for finding the solution.

However, a different approach may be easier to use. Instead of determining $\mathbf{M}_k\mathbf{S}_k$, one can determine $\mathbf{M}_k$ and calculate the other quantities by multiplication. $\mathbf{M}_k$ can simply be determined by solving the following systems:

$$\mathbf{M}_k^{-1}\mathbf{x}_i = \text{column}_i(\mathbf{b}), \tag{6.59}$$

and reconstructing $\mathbf{M}_k$ as:

$$\mathbf{M}_k^{-1} = \sum_{i=1}^{rk} \mathbf{x}_i \left(\text{column}_i(\mathbf{b})\right)^T. \tag{6.60}$$

All the other quantities in the coefficients of equation 6.30 can now be determined.

6.4.5 The calculation of the reference member acceleration

In order to determine the reference member acceleration, equation 6.30 must be solved. It is repeated here for convenience:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T\mathbf{R}_k\right]\mathbf{a}_r = \left[\sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T\mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right]. \tag{6.61}$$

As mentioned earlier, the coefficient of $\mathbf{a}_r$ in equation 6.30 can always be inverted to solve for the reference member spatial acceleration, but this may not lead to valid results. The effective operational space inertia of the system at the reference member is the sum of the inertia of the reference member and the inertias of the chains that are projected to the reference member:

$$\mathbf{I}_{eff} = \mathbf{I}_r + \sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T\phi_k^c\mathbf{M}_k(\phi_k^c)^T\,{}^k\mathbf{X}_r. \tag{6.62}$$

If $\mathbf{M}_k^{-1}$ does not have full column rank, its inverse does not exist. The $\mathbf{M}_k$ calculated by other methods, has only correct components in the directions that chain k allows the reference member to move. The others were implicitly set to zero in the calculations. As discussed in the previous section, these components should actually be infinite, but manipulating matrices where some entries are infinite is usually avoided, and therefore we used zeros. When the spatial inertia of the reference member is added, the effective inertia has full column rank. However, some of its components should have been infinite because the reference member is constrained from movement in some directions by the chains. The system cannot be solved as is. An easy way of rectifying this, is to calculate the image of the applied spatial force on the motion space of the reference member, and then solve the linear system with this image as the applied force. The image of the components of the applied force that does not lie in the motion space of the reference member will be zero, and the solution to the new system will be a correct solution for the reference member acceleration. The motion space of the reference member is simply the base for the column space of $\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{R}_k$:

$$\mathrm{b} = base\left(\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{R}_k\right). \tag{6.63}$$

The new applied force is calculated as:

$$\mathrm{f}' = \mathrm{b}\left(\mathrm{b}^T\left[\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{P}_k - \mathrm{b}_r + \mathrm{g}_r\right]\right), \tag{6.64}$$

and then the following system can be solved in order to obtain the correct acceleration of the reference member:

$$\mathbf{I}_{\mathrm{eff}}\mathrm{a}_r = \mathrm{f}'. \tag{6.65}$$

One the reference member accelerations are known, the correct tip forces can be calculated using it.

6.5 Illustrative example

How to apply the simulation algorithm to a simulation problem might not be clear yet. To assist the reader, a simple problem will now be solved step by step. A lot of insight can be gained by having real numerical values for the system variables and seeing how they interact. The problem chosen is simply two similar links suspending a load, as in figure 6.2. This is perhaps the simplest system that can be treated as a simple closed-chain mechanism. It has only two chains, each with only one link. The beauty of this system is that it can also be treated as a single closed-chain mechanism, and the results of this section can thus be verified with the algorithms developed for single closed-chain mechanisms.

6.5.1 Definition of the structure and initial conditions

In order to have a system with an interesting response, we will give the system an initial displacement by rotating the two links 45^0 counterclockwise and studying the response under the influence of gravity. The coordinate axes are defined as in figure 6.3. The properties of the links are now defined.

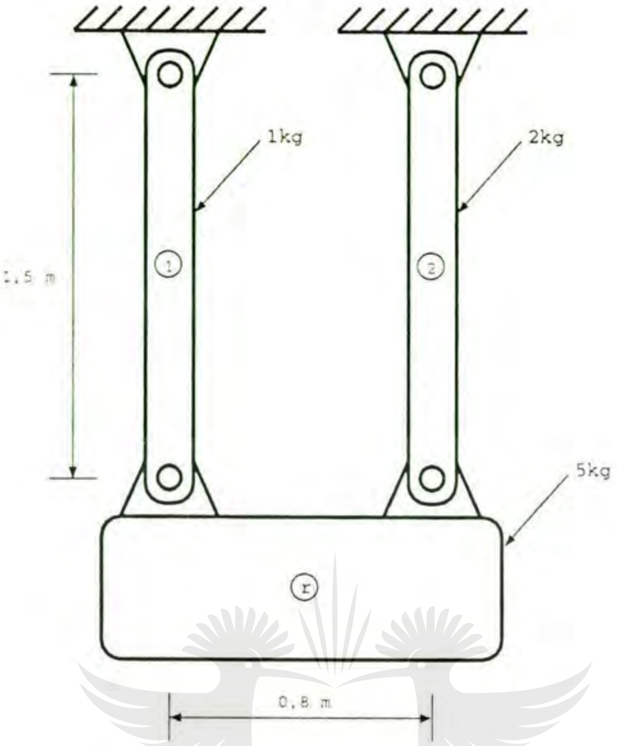


Figure 6.2: A simple closed-chain system.

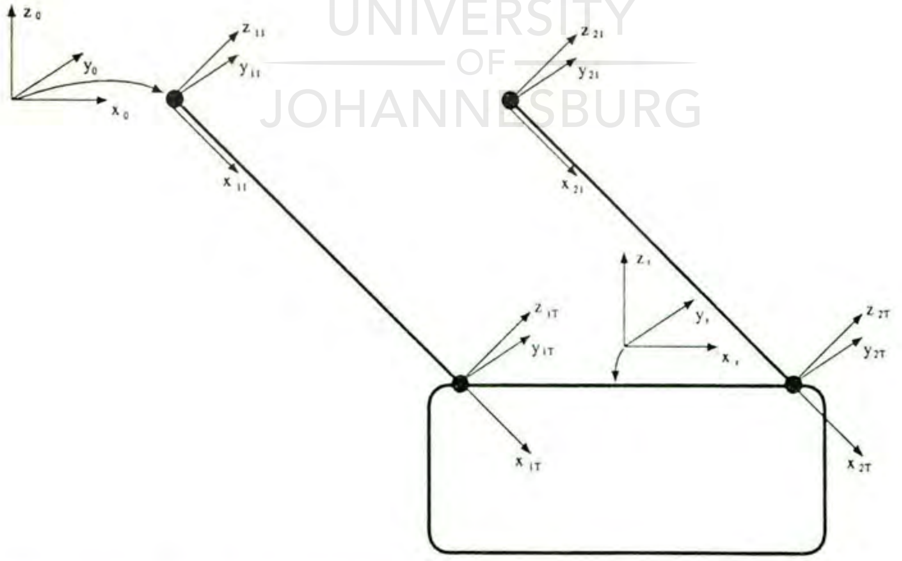


Figure 6.3: The assignment of coordinate frames for the system.

Each link has only one degree of freedom at the interaction with the base, in order to allow rotation of the links in the plane of the paper. The joint space for joint one of both chains[1] is:

$$\phi_{11} = \phi_{21} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6.66}$$

and the constraint space is thus:

$$\phi^c_{k1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.67}$$

We will use the general six degree-of-freedom kinematic notation developed in an earlier chapter. The rotation of the proximal end of the first joint of both chains is taken as zero. The position of the first joints in the base frame, is given as:

$$\mathbf{p}_{11} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6.68}$$

and

$$\mathbf{p}_{21} = \begin{bmatrix} 0,8 \\ 0 \\ 0 \end{bmatrix}, \tag{6.69}$$

respectively. The position of the tip coordinate frames are also described with the general six degree-of-freedom kinematic notation. The resulting joint variables are the same for both chains and are summarized in table 6.1.

Both chains have the following dynamic properties:

$$m_{k1} \;=\; 1\,kg \tag{6.70}$$

[1] The variables and parameters belonging to certain chains will be distinguished from those of other chains by using double subscripting where necessary. The first subscript will always identify the chain, and the second the link. Because of the symmetry of the problem, many quantities will have the same value for both chains. When this happens, the chain will be identified as k, where $k = 1, 2$, implying that the corresponding quantities are identical for both chains.

Joint variable	Joint 1	Tip of chain
$Rot(x)$	0	0
$Rot(y)$	$\pi/4$	0
$Rot(z)$	0	0
$Disp(x)$	1,5	0
$Disp(y)$	0	0
$Disp(z)$	0	0

Table 6.1: The joint variables for both chains (identical).

$$\begin{aligned}
\underline{\mathbf{I}}_{k1} &= \begin{bmatrix} 0,025 & 0 & 0 \\ 0 & 0,25 & 0 \\ 0 & 0 & 0,25 \end{bmatrix} kg\,m^2 && (6.71)\\
\mathbf{s}_{k1} &= \begin{bmatrix} 0,75 \\ 0 \\ 0 \end{bmatrix} m. && (6.72)
\end{aligned}$$

From the above properties, the inertia tensor of each link around the origin of the coordinate frame can be calculated by using the parallel axis theorem:

$$\bar{\mathbf{I}}_i = \underline{\mathbf{I}}_i - m_i \tilde{\mathbf{s}}_i \tilde{\mathbf{s}}_i. \qquad (6.73)$$

The result is:

$$\bar{\mathbf{I}}_{k1} = \begin{bmatrix} 0,025 & 0 & 0 \\ 0 & 0,8125 & 0 \\ 0 & 0 & 0,8125 \end{bmatrix} kg\,m^2. \qquad (6.74)$$

The spatial inertias of the links are calculated from the definition of spatial inertia:

$$\mathbf{I_{k1}} = \begin{bmatrix} 0,025 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,8125 & 0 & 0 & 0 & -0,75 \\ 0 & 0 & 0,8125 & 0 & 0,75 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0,75 & 0 & 1 & 0 \\ 0 & -0,75 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6.75)$$

Now we consider the reference body. Its dynamic properties are as follows:

$$\begin{aligned}
m_r &= 5\,kg && (6.76)\\
\underline{\mathbf{I}}_r &= \begin{bmatrix} 0,5 & 0 & 0 \\ 0 & 0,5 & 0 \\ 0 & 0 & 0,5 \end{bmatrix} kg\,m^2 && (6.77)
\end{aligned}$$

$$\mathbf{s}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} m. \tag{6.78}$$

The inertia tensor around the reference coordinate frame origin is identical to the tensor around the center of mass, since the frame origin coincides with the center of mass. The spatial inertia is simply:

$$\mathbf{I}_{\mathrm{r}} = \begin{bmatrix} 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}. \tag{6.79}$$

The interactions of the two chains with the reference member is defined by the parameters that follow. The only motion allowed between the tip of the chains and the reference member is rotation around one axis, which is an axis parallel to the y-axis of the tip frames (see Figure 6.3). The general joint between the tip of the chains therefore has only one degree of freedom each:

$$(n_c)_k = 1, \tag{6.80}$$

$$\phi_k^c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6.81}$$

$$\phi_k = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.82}$$

The interactions are not completely defined yet. The relative accelerations in the constraint directions of the general tip joints must also be specified:

$$\alpha_k^c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.83}$$

The movements allowed at the interactions of the chains with the reference body member has now been described, but we still have to describe where the chains interact with the reference member. This is described by the position vectors from the origin of the reference member frame to the interaction points:

$$\mathbf{p}_1 = \begin{bmatrix} -0,4 \\ 0 \\ 0 \end{bmatrix}, \tag{6.84}$$

$$\mathbf{p}_2 = \begin{bmatrix} 0,4 \\ 0 \\ 0 \end{bmatrix}. \tag{6.85}$$

The structure and properties of the mechanism has been completely specified. Now the initial conditions and applied forces can be defined. To account for the effect of gravity on the two chains, we give the base frame an acceleration of $10\,ms^{-2}$. The initial joint velocities are all set to zero. In order to study the natural response of the system, all applied joint forces are also set to zero:

$$(\mathbf{\Gamma})_k = [0]. \tag{6.86}$$

The unconstrained forces for the general joint between the tips of the chains and the reference member are also set to zero:

$$\mathbf{h}_k = [0]. \tag{6.87}$$

6.5.2 Open-chain accelerations for the chains

Using the methods described in the section on simulation of single open chains, the Jacobian matrices for the two chains, at their last joints and tips respectively are simply:

$$\mathbf{J_{k,1}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.88}$$

$$\mathbf{J_{k,tip}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ -1,5 \end{bmatrix}. \tag{6.89}$$

(6.90)

The physical interpretation of this is that a unit movement of the first joint (a unit rotation in this case) have the effect of rotating the tip of the chain one unit around the y-axis of the tip frame, and moving the tip one and a half units in the direction opposite to the instantaneous z-axis.

The joint space inertia matrix for both chains can be calculated as:

$$\mathbf{H}_k = [0,8125]\,. \tag{6.91}$$

This means that the moment of inertia of the whole chain (without the reference member) around the axis of joint one is $0,8125\,kg\,m^2$. From this the inverse operational space inertia follows:

$$\begin{aligned}
\mathbf{\Lambda}_{\mathrm{k}}^{-1} &= \mathbf{J}_{\mathrm{k,tip}}\mathbf{H}_{\mathrm{k}}^{-1}\mathbf{J}_{\mathrm{k,tip}} && (6.92)\\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,2307692 & 0 & 0 & 0 & -1,8461538 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1,8461538 & 0 & 0 & 0 & 2,7692308 \end{bmatrix}. && (6.93)
\end{aligned}$$

$\mathbf{\Omega}_k$ also follows:

$$\begin{aligned}
\mathbf{\Omega}_k &= \left(\mathbf{H}_{\mathrm{k}}^{-1}\mathbf{J}_{\mathrm{k}}{}^{T}\right) && (6.94)\\
&= \begin{bmatrix} 0 & 1,2307692 & 0 & 0 & 0 & -1,8461538 \end{bmatrix}. && (6.95)
\end{aligned}$$

The inverse dynamics algorithm, with the joint accelerations set to zero, can be used to determine the bias forces on the chains:

$$\mathbf{C_1}\dot{\mathbf{q}}_1 + \mathbf{G_1} = [-5,3033009]\,. \tag{6.96}$$

In order to balance the effect of gravity on the chains, we would have to apply a moment of about $-5,3\,Nm$ to the first joint of each chain. The open-chain acceleration for both chains can now be calculated because the joint space inertia and applied forces are known:

$$\begin{aligned}
(\ddot{\mathrm{q}}_{\mathrm{open}})_k &= \mathbf{H}_{\mathrm{k}}^{-1}\left[(\mathbf{\Gamma})_{\mathrm{k}} - \mathbf{C}_{\mathrm{k}}\dot{\mathrm{q}}_{\mathrm{k}} + \mathbf{G}_{\mathrm{k}}\right] && (6.97)\\
&= 6,5271395 && (6.98)
\end{aligned}$$

The open-chain operational space acceleration of the tip of the chains can also be calculated. It can be done from the inverse operational space inertia matrix, but then the bias forces in the operational space have to be calculated. It can rather be calculated from the open-chain joint accelerations:

$$\begin{aligned}\ddot{\mathbf{x}}_{\mathbf{k},\mathrm{open}} &= \mathbf{J}_{\mathbf{k},\mathbf{tip}}(\ddot{\mathrm{q}}_{\mathrm{open}})_k + \dot{\mathbf{J}}_{\mathbf{k},\mathbf{tip}}(\dot{\mathrm{q}}_{\mathrm{open}})_k, & (6.99)\\ &= \begin{bmatrix} 0 \\ 6,5271395 \\ 0 \\ 0 \\ 0 \\ -9.7907093 \end{bmatrix}. & (6.100)\end{aligned}$$

6.5.3 The acceleration of the reference member

We have to calculate $\mathbf{S}_k$, $\mathbf{M}_k$, $\mathbf{P}_k$ and $\mathbf{R}_k$, with $k = 1, 2$, before the reference member acceleration can be determined. The bias accelerations of the interaction points, ζ_k, are all zero since the reference member has no spatial velocity at this stage. In the next steps in the simulation, it can have a velocity and these bias accelerations of the interaction point would not be zero. $\mathbf{S}_k$ can be calculated directly from the parameters and variables that has already been calculated or given:

$$\begin{aligned}\mathbf{S}_k &= \left\{\alpha_k^c - (\phi_k^c)^T\left[\zeta_{r,k} + (\ddot{\mathrm{x}}_k)_{\mathrm{open}} + \Lambda_k^{-1}\phi_k \mathrm{h}_k\right]\right\}, & (6.101)\\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -9.7907093 \end{bmatrix}. & (6.102)\end{aligned}$$

$\mathbf{M}_k^{-1}$ can also be calculated from known quantities:

$$\begin{aligned}\mathbf{M}_k^{-1} &= (\phi_k^c)^T \Lambda_k^{-1} (\phi_k^c), & (6.103)\\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.7692308 \end{bmatrix}. & (6.104)\end{aligned}$$

$\mathbf{M}_k^{-1}$ is clearly singular. A base for $\mathbf{M}_k^{-1}$ is:

$$\mathrm{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \qquad (6.105)$$

To determine $\mathbf{M}_k$ for simulation purposed, we solve

$$\mathbf{M}_k^{-1}\mathbf{x} \;=\; \mathbf{b}\,\mathbf{1}_{1\times 1}. \tag{6.106}$$

The particular solution to the above equation is:

$$\mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.3611111 \end{bmatrix}, \tag{6.107}$$

and from this $\mathbf{M}_k$ is constructed as:

$$\begin{aligned} \mathbf{M}_k &= \mathbf{x}\,(\mathbf{b}\,\mathbf{1}_{1\times 1})^T, && (6.108) \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3611111 \end{bmatrix}. && (6.109) \end{aligned}$$

The spatial transformations from the reference member frame to the tip frames of the respective chains can easily be found. The rotation transformation from the reference member frame to the tip frame of chain one can be calculated from equation 6.39:

$$\begin{aligned} {}^1A_r &= \left[({}^{\text{tip}}A_N)_1 \ldots ({}^2A_1)_1({}^1A_0)_1\right]({}^rA_0)^T && (6.110) \\ &= \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0,707107 & 0 & -0,707107 \\ 0 & 1 & 0 \\ 0,707107 & 0 & 0,707107 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \\ &= \begin{bmatrix} 0,7071068 & 0 & -0,7071068 \\ 0 & 1 & 0 \\ 0,7071068 & 0 & 0,7071068 \end{bmatrix} && (6.111) \\ {}^2A_r &= \left[({}^{\text{tip}}A_N)_2 \ldots ({}^2A_1)_2({}^1A_0)_2\right]({}^rA_0)^T && (6.112) \\ &= \begin{bmatrix} 0,7071068 & 0 & -0,7071068 \\ 0 & 1 & 0 \\ 0,7071068 & 0 & 0,7071068 \end{bmatrix}. && (6.113) \end{aligned}$$

The position of the interaction points have been specified before, and the spatial transformations are simply:

$${}^1\mathbf{X}_r \;=\; \begin{bmatrix} {}^1A_r & \mathbf{0} \\ {}^1A_r\tilde{\mathbf{p}}_1^T & {}^1A_r \end{bmatrix} \tag{6.114}$$

$$
= \begin{bmatrix} 0,707 & 0 & -0,707 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0,707 & 0 & 0,707 & 0 & 0 & 0 \\ 0 & -0,283 & 0 & 0,707 & 0 & -0,707 \\ 0 & 0 & -0,4 & 0 & 1 & 0 \\ 0 & 0,283 & 0 & 0,707 & 0 & 0,707 \end{bmatrix} \tag{6.115}
$$

$$
{}^2\mathbf{X}_r = \begin{bmatrix} {}^2A_r & \mathbf{0} \\ {}^2A_r\tilde{\mathrm{p}}_2^T & {}^2A_r \end{bmatrix} \tag{6.116}
$$

$$
= \begin{bmatrix} 0,707 & 0 & -0,707 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0,707 & 0 & 0,707 & 0 & 0 & 0 \\ 0 & 0,283 & 0 & 0,707 & 0 & -0,707 \\ 0 & 0 & 0,4 & 0 & 1 & 0 \\ 0 & -0,283 & 0 & 0,707 & 0 & 0,707 \end{bmatrix}. \tag{6.117}
$$

$\mathbf{P}_k$ and $\mathbf{R}_k$ can now be calculated from known quantities:

$$
\mathbf{P}_k = [\phi_k \mathrm{h}_k + \phi_k^c \mathbf{M}_k \mathbf{S}_k] \tag{6.118}
$$

$$
= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -3,5355339 \end{bmatrix} \tag{6.119}
$$

$$
\mathbf{R}_1 = \left[\phi_1^c \mathbf{M}_1 (\phi_1^c)^{T\,1}\mathbf{X}_r\right] \tag{6.120}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,1021376 & 0 & 0,2553441 & 0 & 0,2553441 \end{bmatrix} \tag{6.121}
$$

$$
\mathbf{R}_2 = \left[\phi_2^c \mathbf{M}_2 (\phi_2^c)^{T\,2}\mathbf{X}_r\right] \tag{6.122}
$$

$$
= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0,1021376 & 0 & 0,2553441 & 0 & 0,2553441 \end{bmatrix} \tag{6.123}
$$

Since the angular velocity of the reference member is zero, by the definition of $\mathbf{b}_r$, $\mathbf{b}_r$ is also zero:

$$\mathbf{b}_r = \left[\begin{array}{c} \omega_r \times \bar{\mathbf{I}}_r \omega_r \\ m_r \omega_r \times (\omega_r \times \mathbf{s}_r \end{array} \right], \quad (6.124)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (6.125)$$

The only external force acting on the reference member is gravity. The center of mass coincides with the origin of the reference frame, therefore gravitational force has no moment around this origin. The spatial external force vector is thus:

$$\mathbf{g}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -50 \end{bmatrix}. \quad (6.126)$$

All the necessary quantities have now been determined and can be substituted into equation 6.30:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T \mathbf{R}_k\right] \mathbf{a}_r = \left[\sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T \mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right] \quad (6.127)$$

$$\left[\mathbf{I}_r + \sum_{k=1}^{m} ({}^k\mathbf{X}_r)^T \mathbf{R}_k\right] = \left(\begin{bmatrix} 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,058 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,361 & 0 & 0,361 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,361 & 0 & 0,361 \end{bmatrix} \right) \quad (6.128)$$

$$= \begin{bmatrix} 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,558 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.361 & 0 & 0,361 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0.361 & 0 & 5,361 \end{bmatrix} . \tag{6.129}$$

$$\left[\sum_{k=1}^{m} ({}^{k}\mathbf{X}_r)^T \mathbf{P}_k - \mathrm{b}_r + \mathrm{g}_r\right] = \left[\begin{bmatrix} 0 \\ 0 \\ 0 \\ -5 \\ 0 \\ -5 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -50 \end{bmatrix}\right] \tag{6.130}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5 \\ 0 \\ -55 \end{bmatrix} \tag{6.131}$$

The system of equations has only the reference member acceleration as unknown and is:

$$\begin{bmatrix} 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5577778 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.3611111 & 0 & 0,3611111 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0,3611111 & 0 & 5,3611111 \end{bmatrix} \mathrm{a}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5 \\ 0 \\ -55 \end{bmatrix} \tag{6.132}$$

Unfortunately $\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{R}_k$ does not have full column rank, and we have to apply the methods discussed in section 6.4.5. We will transform the current system to the system $\mathbf{I}_{\mathrm{eff}}\,\mathrm{a}_r = \mathbf{f}'$ and then solve this system. We first have to calculate the base for $\sum_{k=1}^{m}({}^{k}\mathbf{X}_r)^T\mathbf{R}_k$:

$$\mathrm{b} = base\left(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,058 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,361 & 0 & 0,361 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,361 & 0 & 0,361 \end{bmatrix}\right), \tag{6.133}$$

$$= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0,7071068 \\ 0 & 0 \\ 0 & 0,7071068 \end{bmatrix}. \qquad (6.134)$$

The new applied force follows:

$$\begin{aligned} \mathbf{f}' &= \mathbf{b}\left(\mathbf{b}^T\left[\sum_{k=1}^{m}({}^k\mathbf{X}_r)^T\mathbf{P}_k - \mathbf{b}_r + \mathbf{g}_r\right]\right), & (6.135) \\ &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0,707 \\ 0 & 0 \\ 0 & 0,707 \end{bmatrix}\left(\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,707 & 0 & 0,707 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ -5 \\ 0 \\ -55 \end{bmatrix}\right) \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ -30 \\ 0 \\ -30 \end{bmatrix} & (6.136) \end{aligned}$$

Thus, we want to solve the system

$$\begin{bmatrix} 0,5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,5577778 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.3611111 & 0 & 0,3611111 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0,3611111 & 0 & 5,3611111 \end{bmatrix}\mathbf{a}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -30 \\ 0 \\ -30 \end{bmatrix} \qquad (6.137)$$

The particular solution to this system can be found using any linear system solution technique (like Gauss-Jordan elimination) as:

$$\mathbf{a}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5,2427184 \\ 0 \\ -5,2427184 \end{bmatrix} \qquad (6.138)$$

This is also the sought solution for the acceleration of the reference member (at last). Notice that this acceleration has no rotational components, as one would expect from the symmetry of the problem. Further, the linear acceleration has equal components in the x- and z-directions: the reference member is accelerating with an angle of 45° to the vertical. This direction is exactly tangent to the arc described by the tip of the chains in the current position.

Finding the tip forces for the two chains is easy:

$$\begin{aligned}
\mathbf{f}_k &= \mathbf{P}_k - \mathbf{R}_k \mathbf{a}_r \qquad (6.139)\\
&= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -3{,}536 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \pm 0{,}102 & 0 & 0{,}255 & 0 & 0{,}255 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5{,}2427184 \\ 0 \\ -5{,}2427184 \end{bmatrix}\\
&= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0{,}8581393 \end{bmatrix} \qquad (6.140)
\end{aligned}$$

One can not be sure that this tip force is the real force in the general joint when $\mathbf{M}_k$ does not have full column rank. There might be components of this force that do not lie in the motion space of the tip. These have implicitly been set equal to zero. This has no effect on the calculated motion of the mechanism as a whole, since the motion of the reference member has already been calculated by different means, and the components of the tip force that lies in the constraint directions of the tip has no influence on the motion of that particular chain.

When the relevant components of the tip-force are known, one can treat each chain as a serial chain with a known tip force, and solve for the joint accelerations of each chain. Once again, it is identical for both chains:

$$\begin{aligned}
\ddot{\mathbf{q}}_k &= (\ddot{\mathbf{q}}_{\text{open}})_k - \mathbf{\Omega}_k \mathbf{f}_k \qquad (6.141)\\
&= [6{,}527] - \begin{bmatrix} 0 & 1{,}231 & 0 & 0 & 0 & -1{,}846 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0{,}858 \end{bmatrix}
\end{aligned}$$

$$= [4,9428824] \tag{6.142}$$

The system has now been completely solved. After seeing the amount of effort required to solve a simple system, one may well wonder if this method is worth using. Fortunately, only the open-chain solution might require more effort for larger systems, the rest of the process is similar and requires no extra effort even for large systems.

6.5.4 Verification of the results

There is not much we can do to check the accuracy of the results, except for checking whether the constraints at the interactions of the chains with the reference member are satisfied. The acceleration of the tip of both chains (expressed in the tip frames) are:

$$\begin{aligned} \mathbf{a}_{\mathbf{k,tip}} &= \mathbf{J}_k\ddot{\mathbf{q}}_k & (6.143)\\ &= \begin{bmatrix} 0 \\ 4,9428824 \\ 0 \\ 0 \\ 0 \\ -7,4143235 \end{bmatrix} & (6.144)\end{aligned}$$

The rotation around the local y-axis is not constrained, but the linear motion in the direction of the z-axis is. The motion of the interaction point on the reference member is:

$$\begin{aligned} \mathbf{a}_1 &= {}^1\mathbf{X}_r\mathbf{a}_r & (6.145)\\ &= \begin{bmatrix} 0,707 & 0 & -0,707 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0,707 & 0 & 0,707 & 0 & 0 & 0 \\ 0 & -0,283 & 0 & 0,707 & 0 & -0,707 \\ 0 & 0 & -0,4 & 0 & 1 & 0 \\ 0 & 0,283 & 0 & 0,707 & 0 & 0,707 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5,243 \\ 0 \\ -5,243 \end{bmatrix} & \\ &= -7,4143235 & (6.146)\end{aligned}$$

The constrained components of the above are exactly the same as the corresponding components of the tip acceleration. The constraints are thus satisfied. We can devise another simple test that has to be passed. It is also a necessary but not sufficient condition that this test has to be passed to verify the correct working of the algorithm.

Lets consider the same system with the same initial conditions, except for the applied joint forces at the first joint in the respective chains. We will apply moments such that static equilibrium is maintained. We can easily determine the magnitude of the applied moments necessary to maintain static equilibrium, as shown in Figure 6.4. The result is:

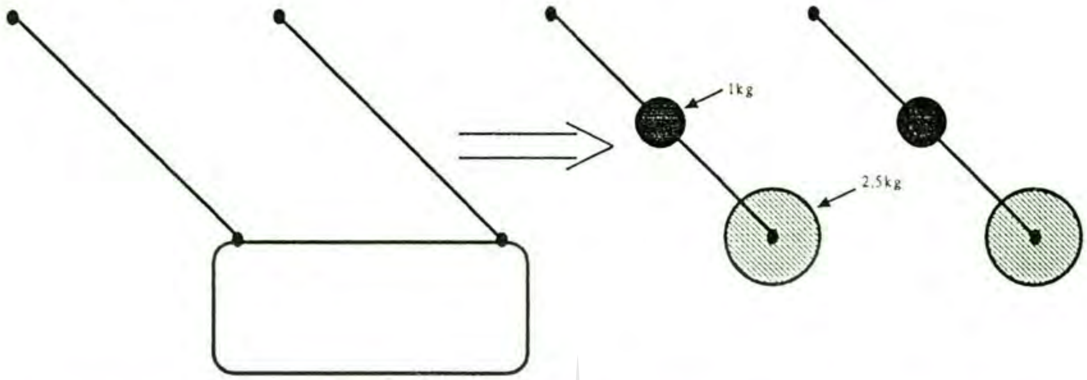


Figure 6.4: An equivalent system when the system is in static equilibrium.

$$
\begin{aligned}
M_y &= -0,75\,m\,\sin\left(\frac{\pi}{4}\right) 10\,N - 1,5\,m\,\sin\left(\frac{\pi}{4}\right)\frac{50}{2}\,N && (6.147)\\
&= -31,819805 && (6.148)
\end{aligned}
$$

This will influence the open-chain joint accelerations and operational space tip accelerations. The new values are:

$$
\begin{aligned}
(\ddot{\mathrm{q}}_{\mathrm{open}})_k &= [-32,635698], && (6.149)\\
(\ddot{\mathrm{x}}_{\mathrm{open}})_k &= \begin{bmatrix} 0 \\ -32,635698 \\ 0 \\ 0 \\ 0 \\ 48,953546 \end{bmatrix}. && (6.150)
\end{aligned}
$$

and if all the necessary calculations are repeated, the acceleration of the reference member is found as:

$$
\mathrm{a}_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \qquad (6.151)
$$

The joint accelerations for the two chains are then:

$$\ddot{\mathrm{q}}_k = [0]. \tag{6.152}$$

The whole system is thus in static equilibrium, just as it should be. This should be quite a relief to all of us who were a bit uncomfortable with all the wild numeric tricks we had to use to solve this system with the given simulation algorithm.

6.6 Summary

In practical applications, many robotic mechanisms can be treated as simple closed-chain mechanisms. Simulating the behaviour of simple closed-chain mechanisms is an even greater challenge than serial closed chains, although the results for serial closed-chains can be used to great benefit in the development of a simulation algorithm for simple closed-chain mechanisms.

How such an algorithm was developed by Lilly and Orin [31] was summarized in this chapter. The basic idea is to identify a central body whose removal will break all closed chains in the system, called the reference member, and to project the inertial properties of all the chains to the reference member. The forces acting on the reference member can be calculated (although not in a trivial way), and since the combined inertia of the system at the reference member has been calculated, the acceleration of the reference member can also be calculated. Once this acceleration is known, the forces exerted by each chain on the reference member can now also be calculated, and the chains can effectively be decoupled from the reference member. These chains can now be treated as serial chains with known tip-forces. The joint accelerations for such systems has previously been partitioned into open-chain and tip-force terms, and these results are simply applied to solve for the joint accelerations of each chain.

Some issues regarding the implementation of these algorithms have been addressed, and the techniques used were illustrated with a worked example.

Chapter 7

The structure of a simulation

Simulating the direct dynamics of a complex simple closed chain mechanism involves many different functions. In this chapter the interaction of these functions will be briefly explained.

7.1 Functions

A functional breakdown of the simulation problem results in the identification of all functions needed to simulate a system. The main functions the problem is split into is the inverse dynamic solution and the direct dynamic solution. These functions need a few supporting functions as well. The more important functions are listed below, with the name of the Scilab implementation in brackets:

- Simple closed chain solution (*multiple_closed_chain*).
- Direct dynamics solution for single serial open or closed chain (*dir_dyn_cor*).
- Inverse dynamic solution for single serial chain (*inv_dyn_friction*).
- Calculation of rotation transformations and relative position vectors from joint parameters (*rot_disp* for Denavit-Hartenberg parameters and *rot_disp6dof* for SDOF parameters).
- Calculation of spatial inertias from the inertia tensors around link centers of gravity (*inertia*).
- Calculation of current joint space if multiple degree of freedom rotational joints are used (*rot_disp6dof*).
- Calculate the rotation from link frame N to the tip frame (*orientation*).

- Calculate the corrective force to be added to the tip force to minimize the effects of integration error (*correction_force*).
- Calculate the positions of the joints relative to the base and sketch the current configuration of the manipulator (*man_plot*).
- Calculate the next state (*next_state*).
- Calculate the joint variables for the next state (*next_joint_var*).

The updating of the joint variables becomes a trivial task when the SDOF kinematic notation is used. The respective joint positions are multipied by the joint spaces, and the non-zero components replace the corresponding components of the joint parameters.

7.2 Structure

The structure of a typical simulation would be similar to that shown in Figure 7.1. The dotted lines indicate paths that are only used for the first time step in the simulation.

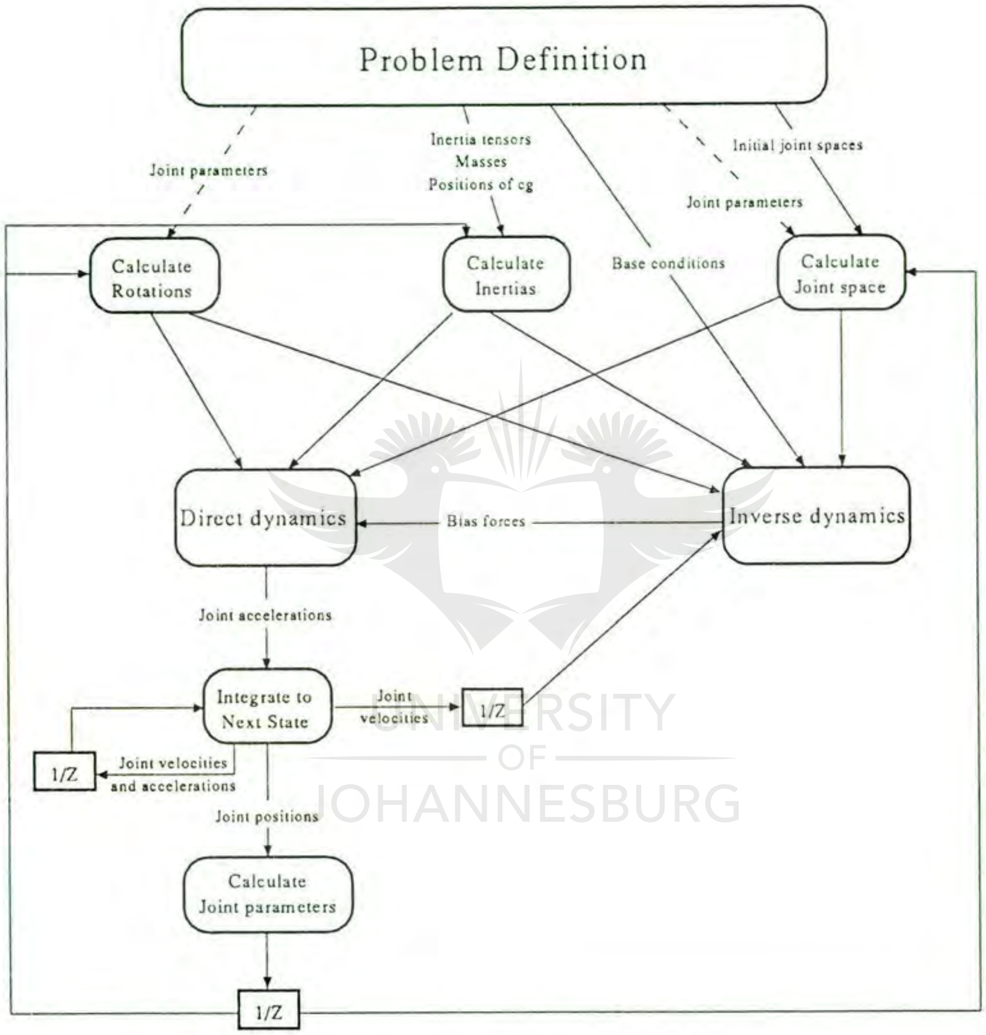


Figure 7.1: The interaction of the different functions in a typical simulation program.

Chapter 8

Complex example

In this chapter the solution to a slightly more complex system than considered in the examples used in previous chapters will be given. The mechanism considered is shown in Figure 8.1. It can be treated as a simple closed-chain mechanism, and contains three serial chains connected by a common reference member. This mechanism has been proposed for use in manufacturing, where the reference member can contain a machine tool like a milling head. The closed-chain configuration of this mechanism gives it superior stiffness and much lower mass – compared to an anthropoid manipulator. As such, this mechanism has been coined a "High Stiffness Low Mass" mechanism.

At first glance it should be clear that the effective control of such a mechanism is not a trivial task. For this reason, a dynamic model that can be used for the simulation of this mechanism, can be most useful in the development and testing of a control system.

8.1 Problem definition

8.1.1 Structure

The default configuration of this mechanism is shown in Figure 8.1. There are now two ways of defining each chain. Each chain can be seen as consisting of two links, the tip of the second link being rigidly attached to the reference member. Alternatively, each chain can be seen as containing only one link, with the tip of this single link being attached to the reference member by a joint. In this example, the first alternative will be used.

The first joint of each of the chains, consists of a spherical joint (two rotational

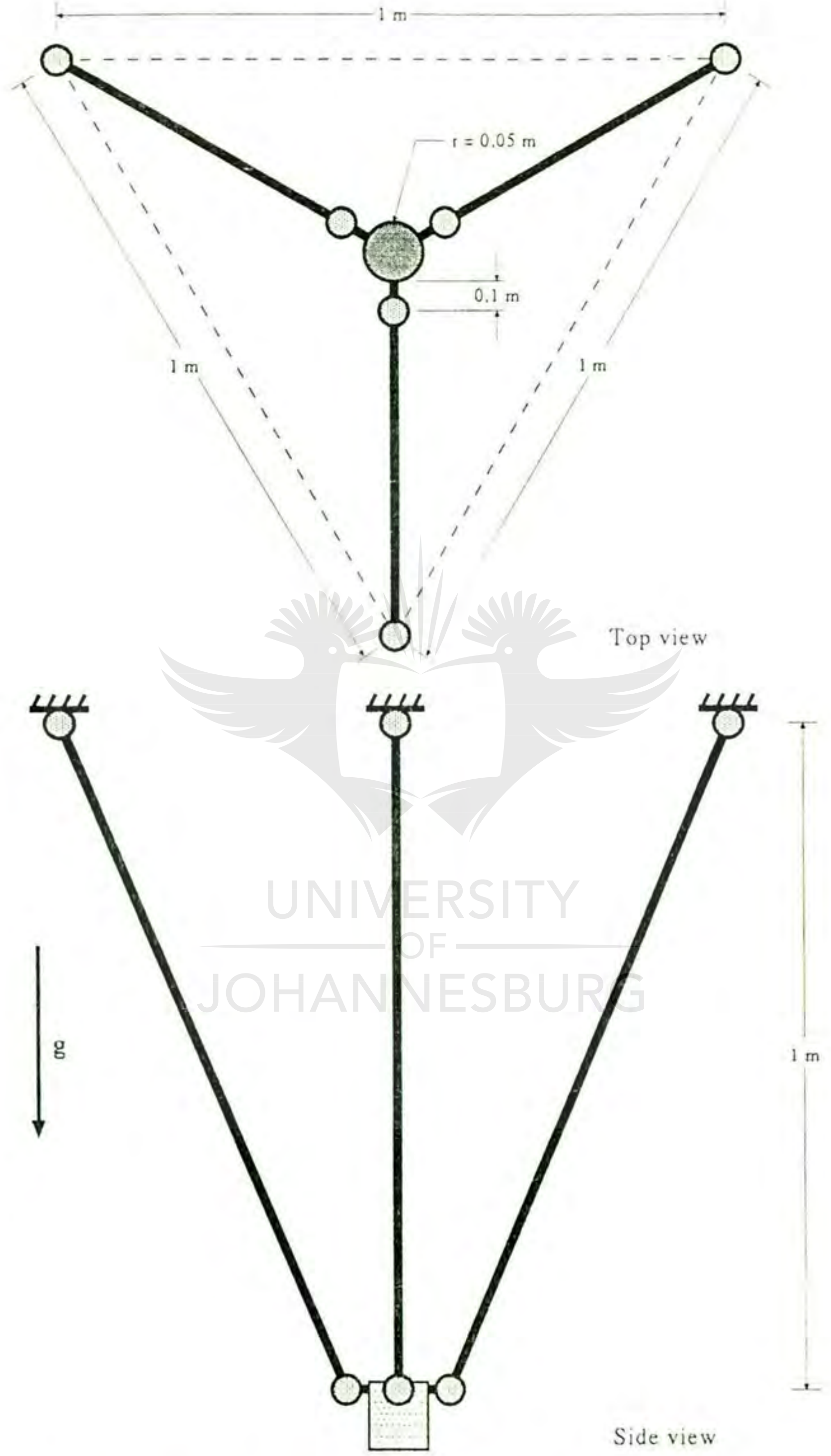


Figure 8.1: A High Stiffness Low Mass Mechanism.

degrees of freedom) and a prismatic joint (one displacement degree of freedom). The second joint of each chain is simply a spherical joint (two rotational degrees of freedom). Each chain thus has five degrees of freedom. This mechanism can reach any point in space, but not with any orientation. This case is more difficult to solve than if each chain had six or more degrees of freedom.[1] The link frames are attached to the links as in Figure 8.2. With the exception of the second component of the joint parameters of link one, the kinematic parameters are the same for each chain. The joint spaces are the same for each chain. The kinematic parameters and joint spaces for the first chain are given in Table 8.1. The dynamic properties of all the chains are also the same. These properties are given in Table 8.2. In the simulation, it is

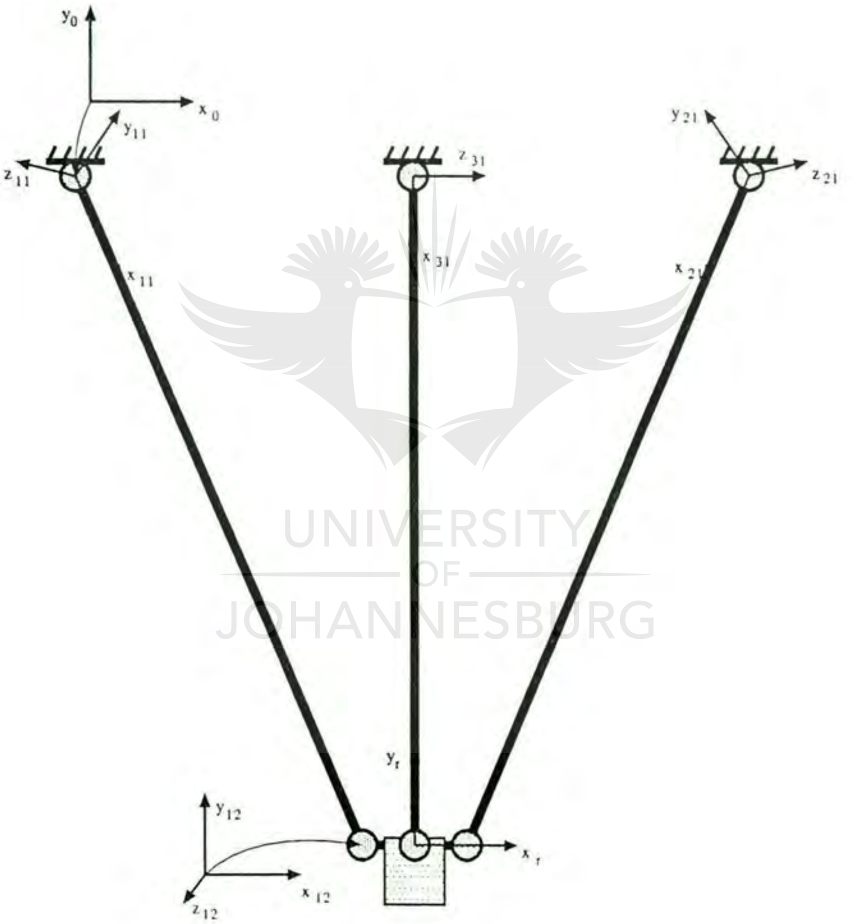


Figure 8.2: The assignment of frames to the "High Stiffness Low Mass" mechanism.'

[1] The operational space inertia, and M_k is not uniquely defined.

Link i	Joint parameters γ_i	Joint space ϕ_i
1	$\begin{bmatrix} 0 \\ -0,5235988 \\ -1,1669367 \\ 1.0874871 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 \\ 0 \\ 1,1669367 \\ 0,1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 00 \\ 10 \\ 01 \\ 00 \\ 00 \\ 00 \end{bmatrix}$

Table 8.1: The SDOF joint parameters and joint spaces of the "High Stiffness Low Mass" manipulator.

Link	Mass	Position of center of mass	Inertia Tensor
1	2	$\begin{bmatrix} 0,5 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,05 & 0 & 0 \\ 0 & 0,5 & 0 \\ 0 & 0 & 0,5 \end{bmatrix}$
2	0,5	$\begin{bmatrix} 0,05 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,00006 & 0 & 0 \\ 0 & 0,0006 & 0 \\ 0 & 0 & 0,0006 \end{bmatrix}$

Table 8.2: The inertial properties of the links of any chain of the "High Stiffness Low Mass" mechanism.

assumed that the position of the center of mass of link 1 stays constant. This is not a bad assumption, if a hydraulic actuator is used, and the prismatic displacements are small – as in this example. The effect on the accuracy of the results are minimal. However, there is nothing to prevent one from assigning an arbitrary function to this position.

The positions of the interactions of the respective chains with the reference member, expressed in the reference member frame, are:

$$\begin{aligned} \mathbf{p}_1 &= \begin{bmatrix} -0,05 \\ 0 \\ -0,025 \end{bmatrix} & (8.1) \\ \mathbf{p}_2 &= \begin{bmatrix} 0.05 \\ 0 \\ -0.025 \end{bmatrix} & (8.2) \\ \mathbf{p}_3 &= \begin{bmatrix} 0 \\ 0 \\ 0.05 \end{bmatrix}. & (8.3) \end{aligned}$$

Many details have not been defined above, as this is not a worked example, but they should implicitly be clear from the information given above.

8.1.2 Control

For simplicity, the control system used is a simple PID-Controller. The only joints that are controlled are the prismatic joint and the second rotational joint of the second link of each chain. The controller determines the applied joint forces in these joints, given the position of these joints. The gains[2] used in the controller are:

$$\begin{aligned} K_P &= 800 & (8.4) \\ K_I &= 1000 & (8.5) \\ K_D &= 50, & (8.6) \end{aligned}$$

and are the same for the prismatic and rotational joints.

[2] These gains were not determined, but chosen.

8.2 Results

8.2.1 Preliminary checks

Before any simulation was done, the function of the algorithms were tested by determining solutions for the following cases [3]:

1. No joint forces applied.

2. Analytically determined joint forces for static equilibrium applied.

Gravitational acceleration was taken as $10\,m/s^2$ – to simplify the interpretation of results. For the first case, the application of the algorithms returned a downward acceleration of $10,0286429\,m/s^2$, while the force exerted on the reference member by each chain included a downward force of $0,0477382\,N$. Each chain had identical joint accelerations of:

$$\ddot{\mathrm{q}} = \begin{bmatrix} 0 \\ -3,6239142 \\ 9,2218498 \\ 0 \\ 3,6239142 \end{bmatrix}. \tag{8.7}$$

It is easy to verify that the above are all consistent. The fact that the reference member has an acceleration greater than gravity might seem strange, but is logical. If the reference member is removed, each of the chains is an open-chain, and the tips of each of these chains will have an acceleration with a downward component greater than gravity.

Consider the second case. A set of analytically determined joint forces are:

$$\Gamma = \begin{bmatrix} 0 \\ 6,3164555 \\ -41,113563 \\ 0 \\ 0,25 \end{bmatrix}. \tag{8.8}$$

For these forces, the algorithms returned no accelerations: the mechanism is indeed in static equilibrium. The algorithms thus passed these necessary tests.

8.2.2 Simulation results

Two simulations were performed. The first simulation involved the following: the mechanism was kept in the default position by a support – no joint forces were applied – and a new controller (integrated error equal to zero) was switched on. The

[3] No controllers used.

command input for the controller was to keep the mechanism in this position. At $t = 0\,s$ the support was instantaneously removed. The simulated response is shown in Figure 8.3. As expected, the reference member falls at first - because of the strong bias in the system and no integrated error - until the displacement (and velocity) is large enough to result in a large force due to the proportional (and derivative) loop. The mechanism does not fall further, but it takes some time for the integrated error to grow enough to bring the mechanism back to its original position. Because of the symmetry, the joint forces are identical for each of the three chains, and are also shown in Figure 8.3.

The second simulation involved taking the above system with the mechanism kept in the original position by the force signals from the controller, and telling the controller to move the reference member $0,25\,m$ vertically upward. The simulated response is shown in Figure 8.4. There is some overshoot, but the controller proves successful.

It looks as if the control forces might diverge, but this is a result of the required kinematic that corresponds to the required position of the reference member not being accurately enough specified, leading to slight contradictions.

Only the prismatic joints of the mechanism needs to be controlled in this mechanism. All the other joint positions depend on these positions. A controller that only controls the prismatic joint forces has been implemented. The gains used are the same as previously. For the same required change in position as for the previous case, the results are shown in Figure 8.5. The control forces no longer diverge, since there can be no contradictions. This controller seems to work well.

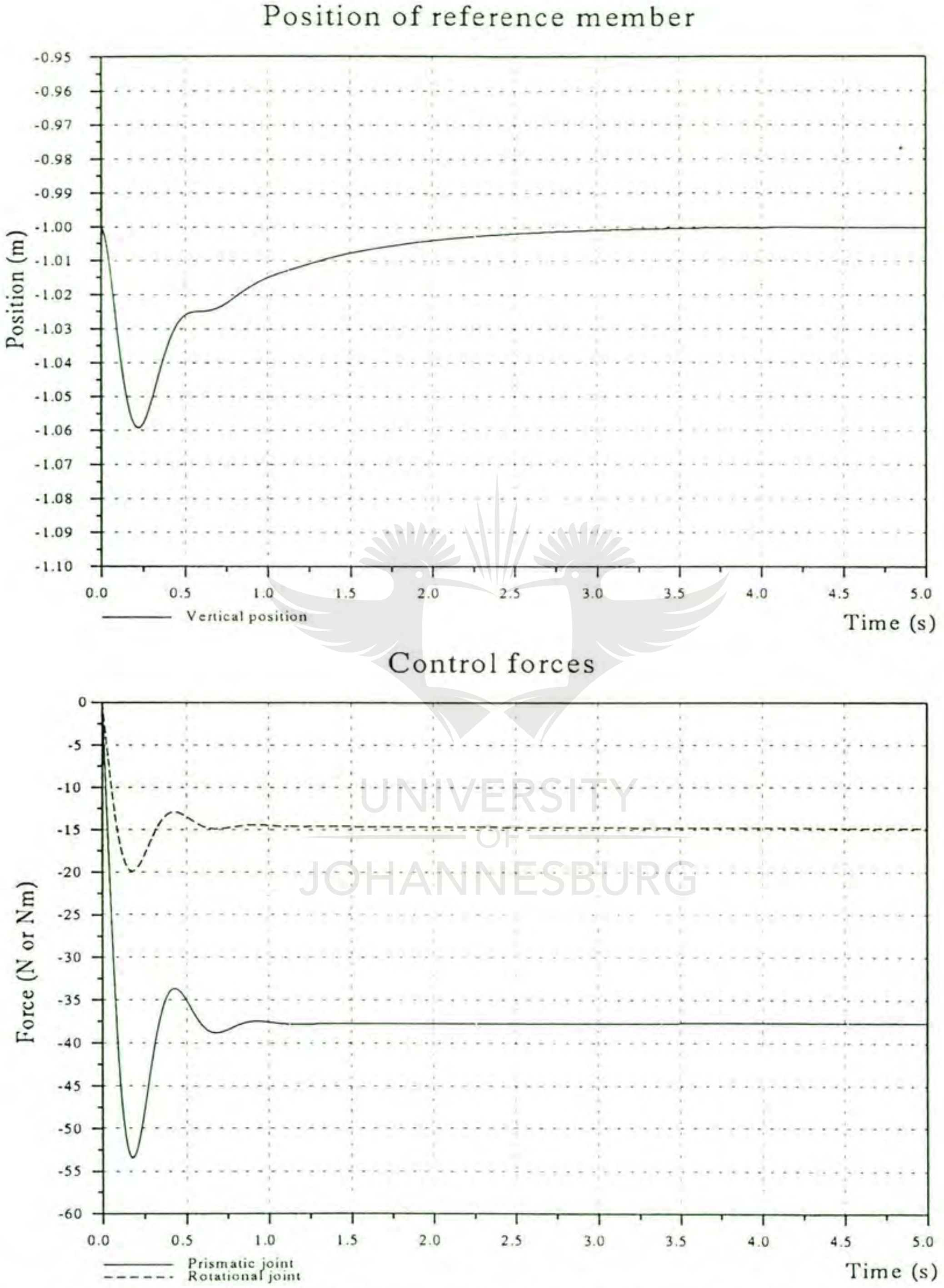
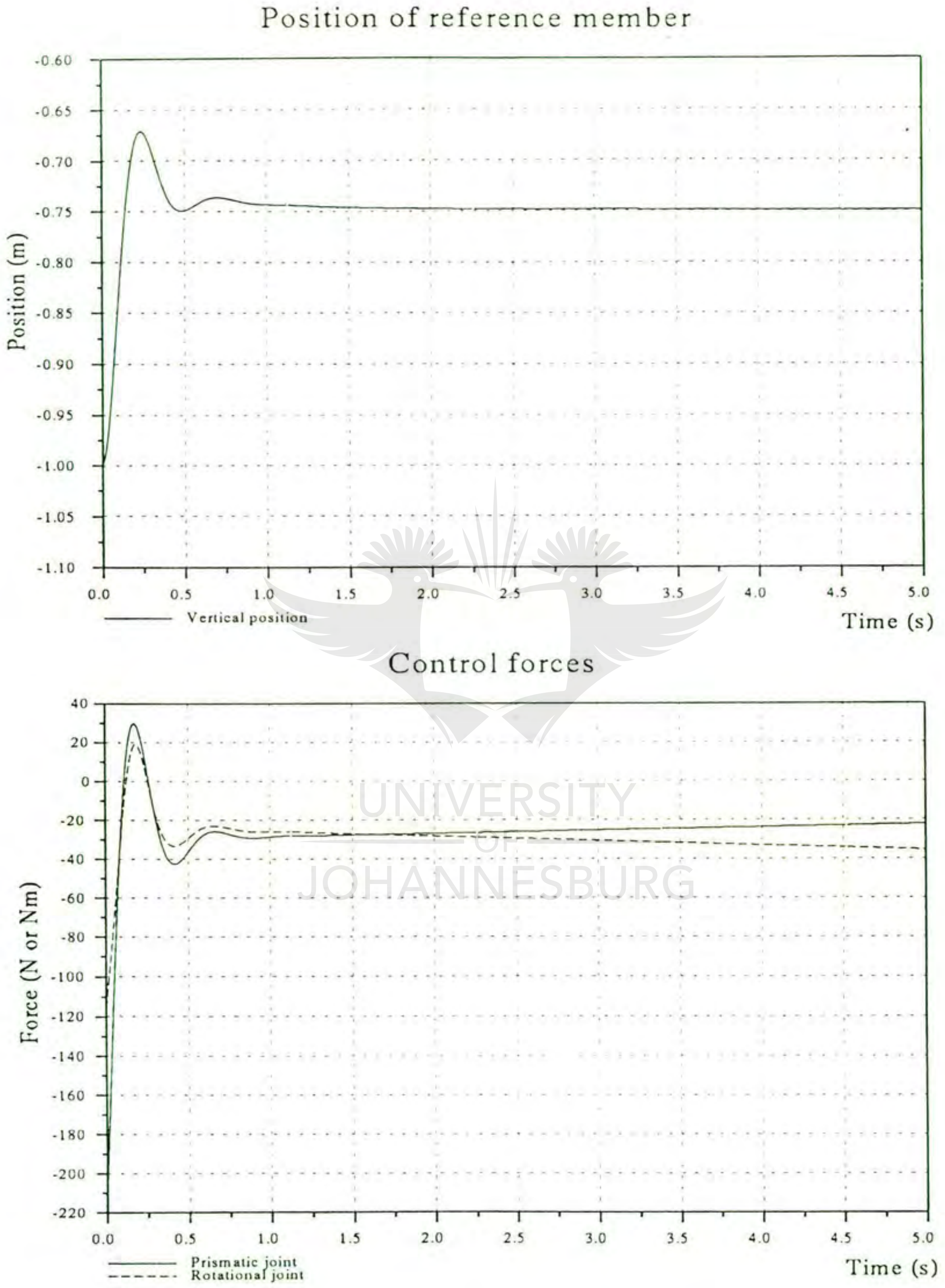


Figure 8.3: The response of the controlled mechanism while trying to keep the mechanism in the original position.



Position of reference member

Control forces

Figure 8.4: The response of the controlled mechanism when commanded to move $0,25\,m$ upward.

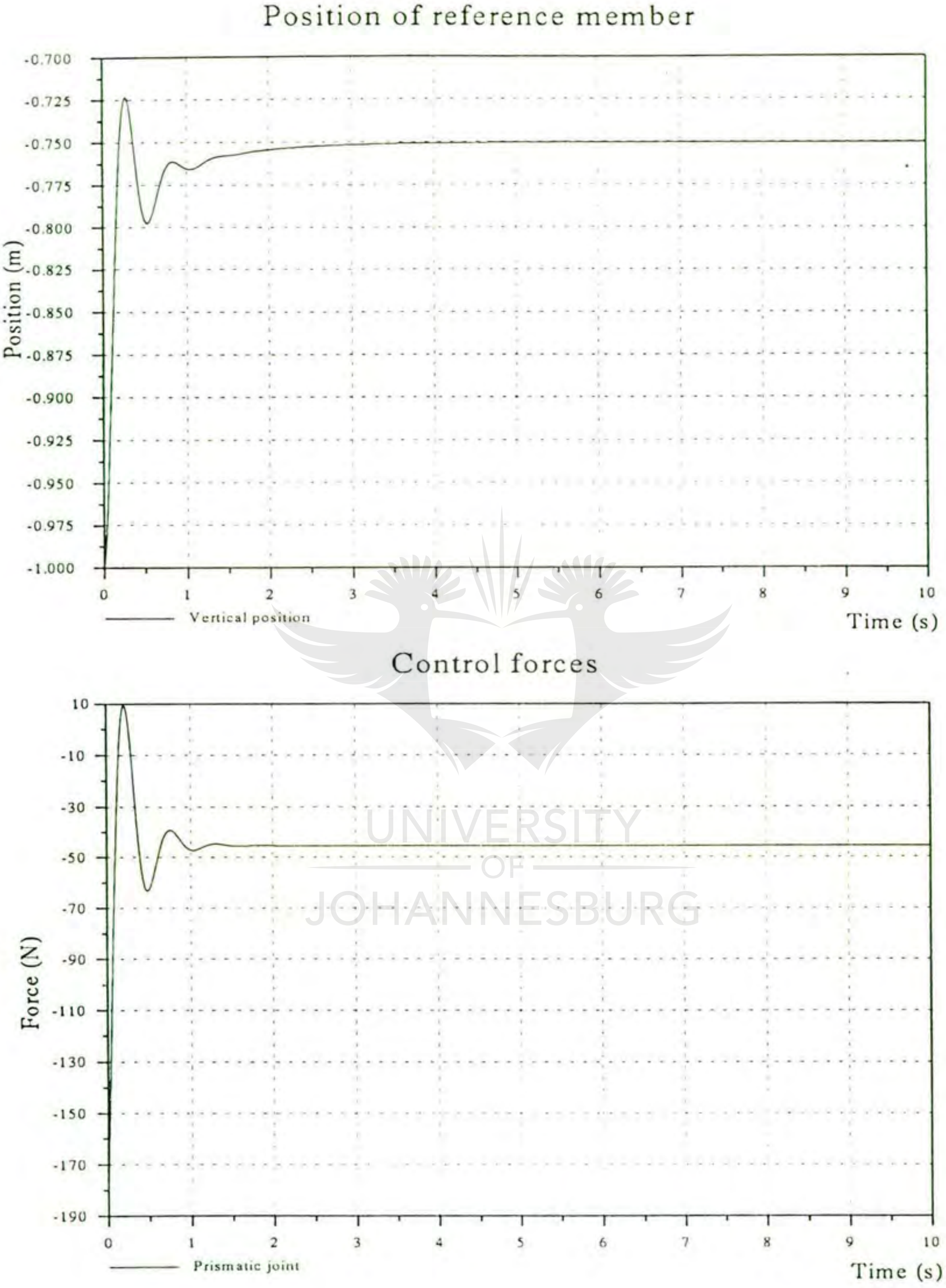


Figure 8.5: The response of the mechanism with only the prismatic joints controlled.

Chapter 9

Conclusion

9.1 Conclusions

In this dissertation compatible methods for modeling the dynamic behaviour of robot like mechanisms has been shown and integrated. Spatial notation and a general joint model was used in both the inverse and direct dynamic models. An inverse dynamic algorithm was derived that is both simple and can handle multiple degree of freedom joints. The direct dynamics algorithms shown and discussed are based on those proposed by Lilly and Orin. A new kinematic notation has been proposed for use with such systems where multiple degree of freedom joints exist. It has been found to be very easy to apply and is conceptually simple – and eliminates the need for massless links when joints with more than one degree of freedom has to be modeled.

The methods mentioned above can be (and have been) used together to simulate relatively complex systems included in the simple closed-chain mechanism with multiple degree of freedom joints class. This includes all single serial closed-chain mechanisms (with less than, equal to or more than six degrees of freedom), and many multiple closed-chain mechanisms. These algorithms can all be used in a simple extension to include other branched structures as well.

The implementation of the dynamics algorithms have also been discussed. This included such details as the origin and handling of singularities in the reference member (of simple closed chain mechanisms) dynamic equation, the calculation of the next state of the reference member, the calculation of the time derivative of the Jacobian, and the functional breakdown of a simulation problem, amongst others. Most of the methods in this dissertation have been illustrated with simple worked examples. All the methods have been coded for a digital computer and used in many simulations, and visualized with simple animations.

Any engineer should be able to use the methods described in this thesis to model rather complex mechanisms – for example by solving the inverse dynamics during the design of new mechanisms, or simulation for the testing of control systems.

9.2 Further work

All the functions needed for the simulation of manipulators have been written and tested. The development of a user friendly graphical front end for these functions is proposed. It should provide a simple way of defining the system and its properties, without sacrificing accuracy or generality. It could be linked to a solid modeler and an animator to show the movements of the mechanism. Ultimately, this can be part of a larger virtual world.

Bibliography

[1] C. Truesdell. *A First Course in Rational Continuum Mechanics*, volume 1. Academic Press, 1977.

[2] K.W. Lilly. *Efficient Dynamic Simulation of Robotic Mechanisms.* Kluwer Academic Publishers, 1993.

[3] J. Wittenburg. *Dynamics of systems of rigid bodies.* B.G. Teubner, Stuttgart, 1977.

[4] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, 1981.

[5] J. M. Hollerbach. A recursive Lagrangian formulation of manipulator dynamics and a comparitive study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics*, 10(11):730–736, November 1980.

[6] T. R. Kane and D. A. Levison. *Dynamics: Theory and Applications.* McGraw-Hill series in mechanical engineering. McGraw-Hill, 1985.

[7] M. Vukobratović and V. Potkonjak. Two new methods for computer forming of dynamic equations of active mechanisms. *Journal of Mechanism and Machine Theory*, 14(3), 1979.

[8] W. M. Silver. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. *International Journal of Robotics Research*, 1:60–70, Summer 1982.

[9] C.A. Balafoutis and R.V. Patel. *Dynamic Analysis of Robot Manipulators: A Cartesian tensor apparoach.* The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1991.

[10] J.Y.S. Luh, M.W. Walker, and R.P. Paul. On-line computational scheme for mechanical manipulators. *ASME of Dynamic Systems, Measurement and Control*, 102:69–79, 1980.

[11] A.L. Zomaya. *Modelling and Simulation of Robot Manipulators - A parallel processing approach.* World Scientific Publishers, 1992.

[12] E. E. Binder and J. H. Herzog. Distributed computer architecture and fast parallel algorithms in real-tiem robot contr4ol. *IEEE Transaction on Systems, Man, and Cybernetics*, 16(4):543–549, July/August 1986.

[13] A. Fijany, I. Sharf, and G. M.T. D'Eleuterio. Parallel o(log n) algorithms for computation of manipulator forward dynamics. *IEEE Transactions on Robotics and Automation*, 11(3):389–400, June 1995.

[14] C. S. G. Lee and P. R. Chang. Efficient parallel algorithm for robot inverse dynamics computation. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(4):532–542, July/August 1986.

[15] H. Anton. *Elementary Linear Algebra.* John Wiley & Sons, Inc., sixth edition, 1991.

[16] J. Steulpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430, October 1964.

[17] R.E. Roberson and R.F. Schwertassek. *Introduction to the Dynamics of Multibody Systems.* Springer-Verlag, Berlin, 1987.

[18] H. Brandl, R. Johanni, and M. Otter. A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix. In *Proceedings of the IFAC/IFIP/IMACs International Symposium on the Theory of Robotics*, pages 95–100, Vienna, Austria, December 1986.

[19] J. Denavit and R.S. Hartenberg. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22:215–221, June 1955.

[20] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research*, 2(1):13–30, Spring 1983.

[21] R. Featherstone. *Robot Dynamics Algorithms.* Kluwer Academic Press, Boston, MA, 1987.

[22] INRIA. *Ψlab 2.1.1*, 1995.

[23] M. Vukobratović and M. Kirćanski. *Scientific Fundamentals of Robotics 4: Real-Time Dynamics of Manipulation Robots.* Springer-Verlag, 1985.

[24] M. Vukobratović, B. Borovac, D. Surla, and D. Stokić. *Biped Locomotion - Dynamics, Stability, control and Application.* Springer-Verlag, Berlin, 1990.

Appendix A

Glossary of terms

Inertial properties The mass, position of the center of mass, and inertia tensor of a body.

Body The mathematical abstraction of the properties of a component (member) of a system in mechanics. This can include the geometrical, inertial and elastic properties of a component.

Link A rigid part in a robotic system between successive joints. It connects successive joints.

Hinge Also called a joint. See the definition of joint.

Joint A structure that connects two links and constrains the relative motion of the two links. Often an actuator is contained in this structure that can exert equal but opposite forces on the two links that the joint connects.

End-effector The part of a robot manipulator that interacts with the environment in such a way as to influence it. It is frequently a gripper or tool. In a serial chain, it is usually the far end of the last link in the chain.

Non-inertial frame A frame of reference in which the Newton-Euler laws of motion appear not to be valid if the variables are measured in this frame. A frame that has an acceleration or angular velocity relative to an inertial frame, will be a non-inertial frame.

Proper transformation A continuous mapping such that the inverse image of a compact set is compact.

Skew-symmetric tensor A tensor $\mathbf{T}$ such that $\mathbf{T}^T = -\mathbf{T}$.

Tensor A multilinear differential form with respect to a group off permissible coordinate transformations in n-space; an element of a tensor product. A tensor of type (r, s) is a member of the product

$$\mathbf{T}^r_s = \mathbf{T}^* \odot \ldots \odot \mathbf{T}^* \odot \mathbf{T} \odot \ldots \mathbf{T} \quad . \qquad (A.1)$$

of the vector space $\mathbf{T}$ with itself r times and with its dual, $\mathbf{T}^*$, s times. Different bases of $\mathbf{T}$ lead to different bases of $\mathbf{T}^*$, and hence to different bases for the tensors; however, a component transformation law exists. If we choose the basis of $\mathbf{T}$ to be orthonormal, we obtain Cartesian tensors. A *zero-order tensor* is a scalar and has no superscript or subscript. A *first-order tensor* is a member of $\mathbf{T}$ or $\mathbf{T}^*$ (according to whether it is covariant or contravariant), and corresponds to a vector; it has one subscript or superscript. A *second-order tensor* can be represented by a matrix, and the total of its subscript and superscript is two, that is, it is a member of $\mathbf{T}^2$, $\mathbf{T}_2$, or$\mathbf{T}^1_1$. In this thesis, the subscripts and superscripts are not used as from the context it should be clear whether second-order Cartesian tensors, vectors, or scalars are implied.

Elementary transformation A coordinate frame transformation in which one of the axis does not change orientation.

Base A body in space whose motion is known and not influenced by the action of other bodies on it. It is usually an immovable object to which a part of a manipulator is fixed.

Jacobian A matrix that relates the joint velocities to the spatial velocity of the end effector.