UNIVERSITA' DEGLI STUDI DI PISA
Dipartimento di Sistemi Elettrici e Automazione
Centro Interdipartimentale di Ricerca "E. Piaggio"

Tesi di Dottorato
Dottorato di Ricerca in Automatica, Robotica e Bioingegneria
Ciclo XXI

STIFFNESS AND COMPLIANCE OF KINEMATIC CHAINS IN MOTION

Tutore:
Prof. Danilo De Rossi

Studente:
Claudia Caudai

To move things is all that mankind can do...for such
the sole executant is muscle, whether in
whispering a syllable or in felling a forest.
Charles Sherington, 1924

Contents

List of Figures

Chapter 1

Introduction

Bioengineering is a very fascinating kind of engineering. It applies engineering principles to the full spectrum of living systems and needs, in its hard challenges, a deep multidisciplinariety between biology, chemistry, physiology, mathematics, physics, informatics, robotic, neuroscience and many other disciplines.

Nature of biological bodies is so surprising, for the complexity, the perfection and the armonicity of its micro and macro structures, for the capacity of adaptation in several hard conditions and for the geniality of tricks adopted in the surviving race.

In this sense Bioengineers need a great understanding of living systems and a great capacity of abstraction and synthesis.

In this thesis I deal with the problem of motor control, with a special attention to the muscle, studied in its biological context, seen as the main instrument of realization of a motor planning and actuation of a particular wish. The relation between the structure of the muscle and its functionality is very deep.

The spring-like muscle behaviour has long been recognized as a key element in the control of movements. The main questions are: which

trajectory should be used from the initial to the desired position? What muscle forces should be generated? In 80s many dynamic optimization approaches have been proposed, based on the idea that, for each motor behavior, the nervous system uses a strategy which minimizes the number of changeable control variables. In 1985 Flash and Hogan proposed the minimum-jerk model. Using this strategy, the nervous system is choosing the (unique) smoothest trajectory for any horizontal movement [22] [27] [38]. Similar strategies were formulated at the end of 80s by Uno et al., the minimum-torque-change model and the minimum-muscle-tension-change model [39] [15]. In 1986 Feldman proposed the λ-model of motor control [16], according to which the motion of a joint arises through a reciprocal command, which alters the relative activation of agonist and antagonist muscles to produce a shift in the equilibrium between the muscle force and external loads. According to the hypothesis of Feldman, intentional movements are produced by the nervous system shifting positional frames of reference for the sensorimotor apparatus. This hypotesis can be extended to multi-muscle and multi-degrees of freedom systems, providing a solution for the redundancy problem and allowing the control of a joint combination to produce any desired limb configuration and movement trajectory.

In my work I adopted the *Feldman Muscle Model* [16], introduced and explained in the first chapter.

The second chapter illustrates the concepts of stiffness and compliance of a motionless kinematic chain.

In the third chapter the definiton of the Dynamic Stiffness Operator is given. It represents a new indicator, introduced by the use of functional analysis, which allows a theoretical and practical study of the performances of a chain during collisions or under external perturbations.

The fourth chapter is devoted to applications. Examples of kinematic

chains with two or three degrees of freedom are controlled in position and stiffness.

In the fifth chapter the definition of the Dynamic Compliance Operator is given. It is defined as the inverse of the Stiffness Operator and is very more complex to calculate explicitly. In this chapter many mathematical instruments to develop a theory, which permits to evaluate an approximation of this operator, are used.

The sixth chapter gives a look out upon the possible applications, in the field of biorobotic, of the theory developed.

Chapter 2

Motion Control Theory

2.1 The BioMechLab Group

My work is contextualized in the activity of the work group *BioMechLab*[1]. The research activity of this group deals with the synthesis and the analysis of human movement. Its main aims are the design and development of innovative methodologies and technologies devoted to the realization of haptic and kinesthetic interfaces able to analyze, code and replicate human movement. The group is constituted by people from the Interdepartmental Center "E.Piaggio", the Department of Information Engineering and the Department of Electrical System and Automation of the University of Pisa. A large part of their activity consists of prototype realization and methods of implementation concerning human movement analysis. The research activity of the group *BioMechLab* is subdivided in four main topics:

- The study of Conductive Elastomers (CE) composites which show piezoresistive properties when a deformation is applied. The CEs

[1] http://biomech.iet.unipi.it/

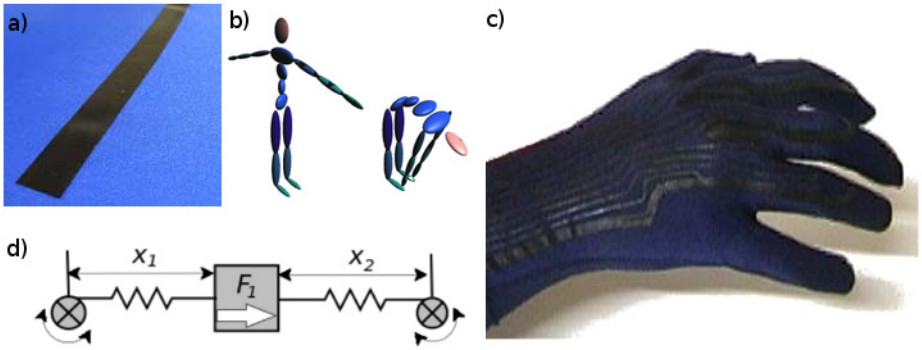

Figure 2.1: *a) Conductive Elastomers (CE). b) User Interfaces. c) Sensing Systems d) Mathematical Models.*

can be integrated into fabric or other flexible substrate to be employed as strain sensors (Fig.2.1 a).

- The realization of biomechanical models of the human body, innovative algorithms and graphical representations, to interpret data deriving from sensing garments (Fig.2.1 b).

- The production of sensing fabrics smeared with elastosil and trichloroethylene on a lycra substrate previously covered by an adhesive mask, according to the desired topology of the sensor network and cut by a laser milling machine (Fig.2.1 c).

- The mathematical analysis of the performances and controllability in compliance an stiffness of kinematic chains constituted by polymeric elastomers actuators, in the static and in the dynamic case (Fig.2.1 d).

The work of my PhD is represented by some specific contributions to the work of the BioMechLab's group mainly regarding the last topic.

2.2 A *Biomimetical* Approach

The word "Biomimetic" was introduced by Otto H. Schmitt in 1969 [32], [33]. It refers to human-made processes, substances, devices, or systems utilized in copying, imitating, and learning from nature and biology. The Biomimetics concern the art and science of designing and building apparatus similar to the biological ones, and is of special interest to researchers in nanotechnology, robotics, artificial intelligence (AI) and the medical industry.

The biomimetic robots represent a new class of robots which will be substantially more compliant and stable than current robots, and will take advantage of new developments in materials, fabrication technologies, sensors and actuators. Adapting mechanisms and capabilities from nature and using scientific approaches led to effective materials, structures, tools, mechanisms, processes, algorithms, methods, systems and many other benefits.

The transfer of technology between life forms and synthetic constructs is desirable because evolutionary pressure typically forces living organisms (fauna and flora) to become highly optimized and efficient.

2.3 The Principles of Movement

Humans and other primates can easily perform a wide variety of tasks without much knowledge about themselves and environment. This contrasts with the current state of robotics: even for a robot to reach to a position with natural pose can be a research topic, much less for the robot to be as dexterous and intelligent as humans.

2.3.1 The Muscles

Animals use muscles to convert the chemical energy of ATP into mechanical work. There are three different kinds of muscles in vertebrate animals:

Smooth muscles are involuntary (they cannot be controlled voluntarily). They are found in the walls of all the hollow organs of the body (except the heart). Their contractions reduce the size of these structures.

Striated muscle are also called *Skeletal muscles* because of their anatomical location, are composed by a large number of muscle fibers, which range in length goes from 1 to 40 mm and in diameter from 0.01 to 0.1 mm. Each fiber forms a (muscle) cell and is distinguished by the presence of alternating dark and light bands. This is the origin of the name "striated"(see Figure 2.2).

Cardiac muscle makes up the wall of the heart. It contracts about 70 times per minute pumping 5 litres of blood each minute.

Skeletal Muscles

Each skeletal muscle fiber is a single cylindrical muscle cell (Figure 2.3). An individual skeletal muscle may be made up of hundreds, or even thou-

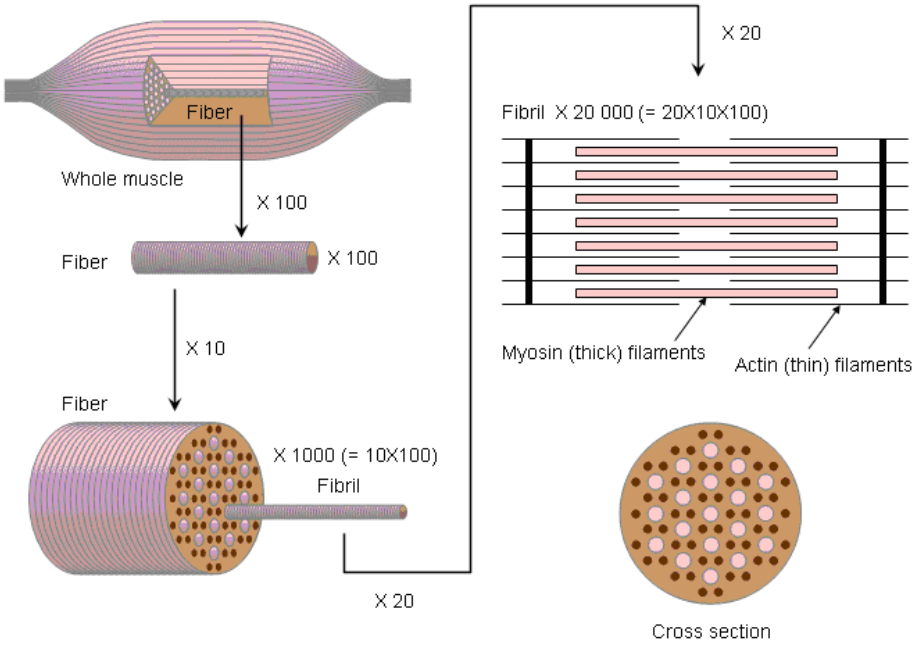

Figure 2.2: *Anatomy of striated muscle. The fundamental physiological unit is the fiber.*

sands, of muscle fibers bundled together and wrapped in a connective tissue covering. Each muscle is surrounded by a connective tissue sheath called *epimysium*. Portions of the epimysium divide the muscle into compartments. Each compartment contains a bundle of muscle fibers. Each bundle of muscle fiber is called *fasciculus* and is surrounded by a layer of connective tissue called *perimysium*. Within the fasciculus, each individual muscle cell is surrounded by connective tissue called *endomysium*. Skeletal muscles act in pairs, the flexing (shortening) of one muscle is balanced by a lengthening (relaxation) of its paired muscle or a group of muscles. Muscles that contract and cause a joint to close are called *flexor muscles*, and those that contract to cause a joint to stretch out are called *extensors*. Skeletal muscles supporting skull, backbone, and rib cage are called *axial skeletal muscles*; whereas, skeletal muscles of the limbs are called *distal*. These muscles attach to bones via strong, thick connective tissue called *tendons*. Several skeletal muscles work in a highly coordinated way (performing complex activities such as walking).

Motor Units

A motor unit is the name given to a single alpha motor neuron and all the muscle fibers it activates. With 250 millions of skeletal muscle fibers in the body and about $420,000$ motor neurons, the average motor neuron is able to stimulate about 600 muscle fibers. Large muscles may have about 2000 fibers per motor unit, while the tiny eye muscles may have only 10 fibers per motor unit. The size of a motor unit varies considerably according to the muscles function. Muscles controlling high precision movements, like those required in the fingers or in the eyes movements, are organized into smaller motor units. The motor unit is the brains smallest force-control functional unit.

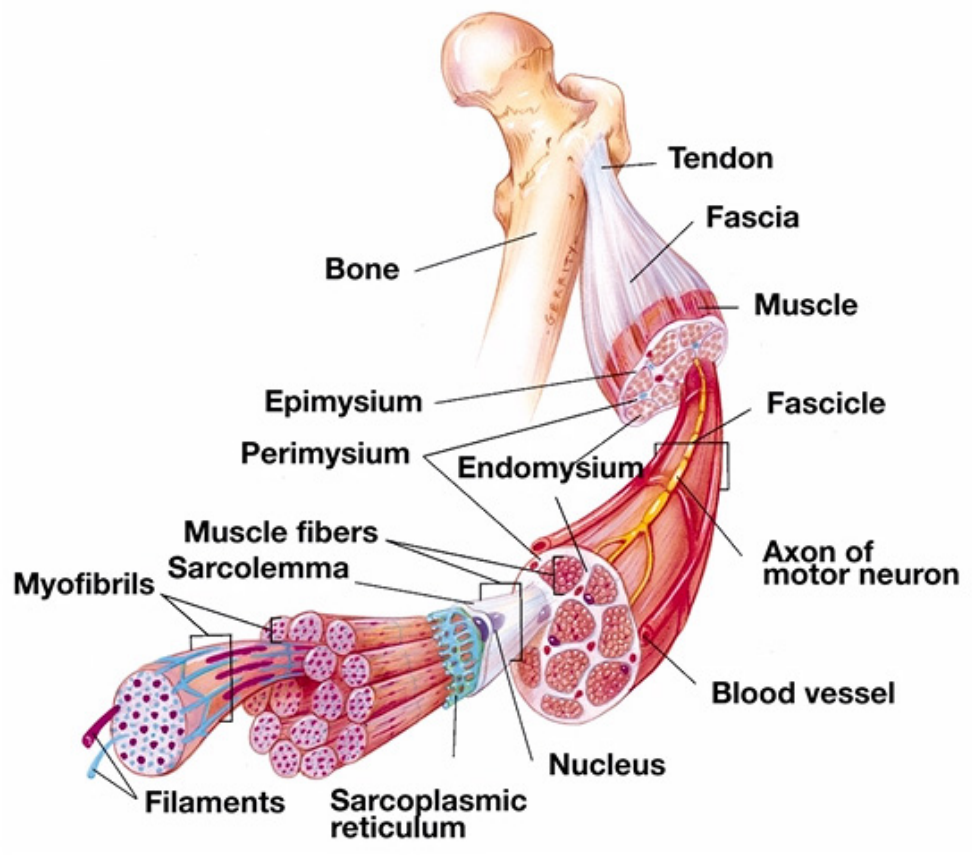

Figure 2.3: *Structure of skeletal muscle.*

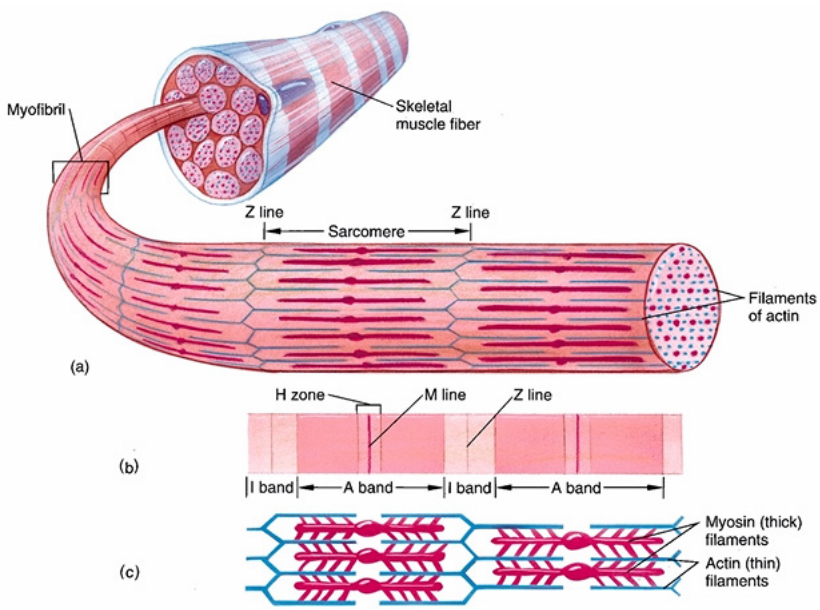

Figure 2.4: *Structure of sarcomere unit.*

Sarcomers

Each skeletal muscle cell has several *myofibrils*, long cylindrical columns of myofilaments, which form the striated sarcomere unit. The thick myosin filaments of the sarcomere provide the dark, striped appearance in striated muscle, and the thin actin filaments provide the lighter sarcomere regions between the dark areas (see Fig.2.4). A sarcomere can induce muscle contraction. The actin and myosin filaments slide one over the other. Muscle contraction creates an enlarged center region in the whole muscle. The flexing of a bicep makes this region anatomically visible. This large center is called the *belly of the muscle*.

Neuromuscular Junctions

The nervous system "communicates" with muscles via neuromuscular junctions. These junctions (Fig. 2.5) work like a synapse between neurons:

1. The impulse arrives at the end bulb.
2. Chemical transmitter is released from vesicles and diffuses across the neuromuscular cleft.
3. The transmitter fill receptor sites in the membrane of the muscle and membrane permeability to sodium increase.
4. sodium diffuses in the membrane and, if the threshold is reached, an action potential occurs.
5. an impulse travels along the muscle cell membrane, and the muscle contracts.

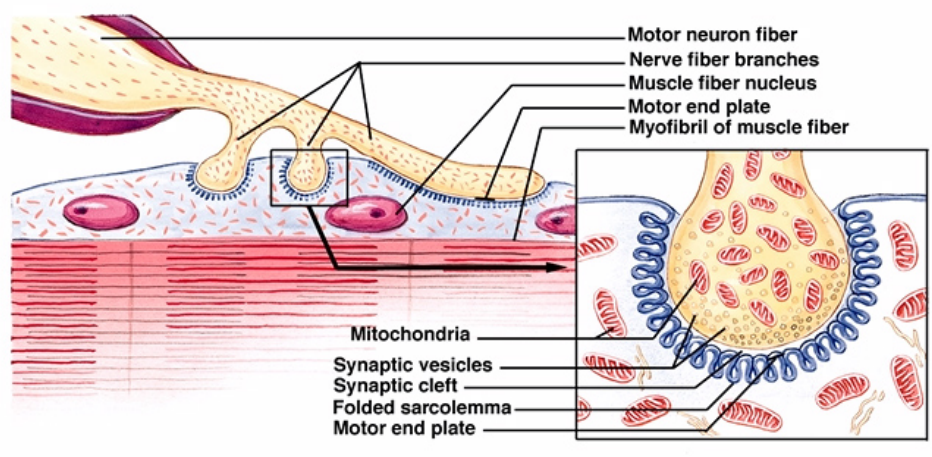

Figure 2.5: *The neuromuscular junction.*

2.3.2 Regulation of Force

The brain combines two control mechanisms to regulate the force of a single muscle.

- The first is *RECRUITMENT*. The motor units that make up a muscle are recruited according to the *Size Principle*. Smaller motor units (fewer muscle fibers) have a small motor neuron and a low threshold for activation. These units are recruited first. As more force is demanded by an activity, progressively larger motor units are recruited. This has great functional significance. When requirements for force are low, but control demands are high (writing, playing the piano) the ability to recruit only a few muscle fibers gives the possibility of fine control. As more force is needed the impact of each new motor unit on total force production becomes greater. The smaller motor units are generally slow units, while the larger motor units are composed of fast twitch fibers.

- The second method is called *RATE CODING*. Within a given motor unit there is a range of firing frequencies. Slow units operate at a lower frequency range than faster units. Within that range, the force generated by a motor unit increases with increasing firing frequency. If an action potential reaches a muscle fiber before it has completely relaxed from a previous impulse, then force summation will occur and the muscle will generate a larger force.

2.3.3 The Biorobotic

The analysis of static and dynamic properties of muscles, concerning the chemical and physical properties of the musculo-skeletal system, can be applied to bio-robotic, in order to design robots following these principles and to implement in robots basic control strategies for compliance, safety, energy efficiency, close to the ones adopted by the biological body. The design of a biomimetic control strategy begins from the knowledge of the musculo-skeletal system behaviour and the observation of interesting intrinsic characteristics related to muscle distribution in actuation terms and to their arrangements within the skeletal system.

In biological systems the redundancy of muscles, in respect to the number of the degrees of freedom, leads the formulation of several control strategies, based on different criteria concerning either muscular synergies empirically proved, the concept of reflex, the minimization of functionals related to mechanical quantities regarding the motion (energy, torque, jerk), the integration of the informations derived from external environment, acquired by the proprioceptive sense, or strategies of self-adaptation.

2.4 An Enlarged Concept of Stiffness

During a movement, when we reach for an object, we choose a simple trajectory, and we perform it with a smooth and stable motion. This probably means that the muscles and the associated spinal reflex circuitry are designed in a way that makes control of motion particularly simple for the brain. The consequence of this observation could be that a well designed muscle-skeletal system can simplify and sometimes solve much of the problems inherent to the control of a kinematic chain.

One of the main aims in motion control theory is to provide a collection of variables defined by the Central Nervous System which allow position-control of the status of a biological kinematic chain. The choice of this set depends on the theory adopted and there is still not a common criteria universally accepted by the scientific community.

2.5 Feldman's Muscle Model

In my PhD work, I took into account the **Equilibrium Point Control Theory** formulated by Anatol G. Feldman in the early 80. Feldman proved a direct connection between the magnitude and the frequency of sub-cortical electrical stimuli and muscular co-activation.

As we have seen in Section 2.3 the activity and the force exerted by a muscle in a biological system is regulated by the Central and the Peripheral Nervous Systems through a twofold mechanism:

1. by means of frequency variations of the electrical activity of motoneurons;

2. by means of an increment of the number of active motoneurons (the *motorneuronal recruitment*).

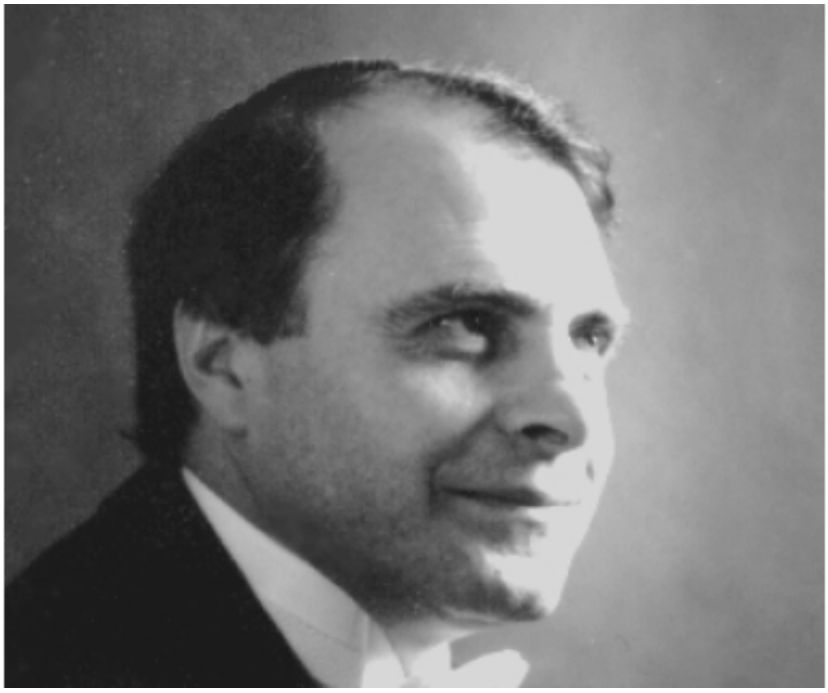

Figure 2.6: *Anatol G. Feldman*

During his experimental trials on decerebrated cats, by stimulating subcortical nervous centers with electrical signals, Feldman proved that muscles can exert different forces according to the frequency of the stimuli by maintaining the same length. He proposed a muscle model based on the scheme of equations (spanned by p):

$$F = \begin{cases} \alpha(x-\lambda)^p & x > \lambda \\ 0 & x \leq \lambda \end{cases} \tag{2.1}$$

where x represents the actual length of the considered muscle, λ the *rest length* (i.e. the maximum length a muscle can assume without performing any force) and α a mechanical parameter.

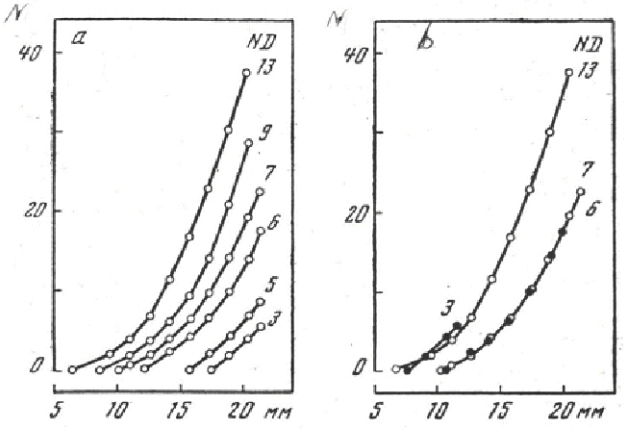

Figure 2.7: *Comparison for shapes of the various mechanical characteristics of the gastrocnemius muscle of decerebrated cat.*

Really, a simple explicit relation among an electrical signal stimulating the muscle and the parameter α was not found. To find an explicit relation Feldman maintained unchanged the electrical stimulus; in this way he discovered that, by varying the length of the muscle, the forces performed was increasing proportionally with the square of the muscle length (see Fig 2.7).

This case is described by Eq. 2.1 by assuming $p = 2$ if α (which is replaced by k, elastic constant) does not practically depend on the stimulus and on the muscle length. In this case Feldman's muscle model becomes:

$$F = \begin{cases} k(x-\lambda)^2 & x > \lambda \\ 0 & x \leq \lambda \end{cases} \tag{2.2}$$

Each characteristic is obtained by unchanging the frequency of the electrical stimulus, represented by a parameter which can be used as a central control. It means that, in constant electrical conditions, exists a unique parameter (λ) characterizing the force exerted by a muscle in

respect to the stretching. This fact can be observed also graphically, in effect each curve in Eq. 2.2 intersects the abscissa axis in a unique point (λ). The parameter λ is called *rest length* because represents the maximum length that a muscle can reach without exerting any force and can be used to label each mechanical characteristic.

In this sense Feldman formulated one of his main results:

The CNS directly controls the
rest length of a muscle.

This means that the force exerted by a muscle is regulated by the CNS, in the sense that once the rest length is decided, the force exerted is proportional to the external loads which produce the actual length x greater than λ.

Chapter 3

Definition of Stiffness and Compliance

In order to explain the concepts of mechanical stiffness and compliance, let us consider a general case of a kinematic chain with n degrees of freedom collected in a geometric state vector $\mathbf{q}$ ($[\mathbf{q}] = n$)(see Fig. 3.3). The chain is actuated by a functional group controlled by a vector λ (whose length m is greater than n). By introducing suitable functions $y_i(\mathbf{q})$ of the state $\mathbf{q}$, it is possible to define a "supplementary" state vector $\mathbf{c}$ (or $\mathbf{s}$), having length p, of generalised compliances C_i (or stiffness S_i)

$$C_i = \left\| \frac{\partial y_i(\mathbf{q})}{\partial P} \right\| \quad S_i = \left\| \left(\frac{\partial y_i(\mathbf{q})}{\partial P} \right)^{-1} \right\| \quad i = 1..p \tag{3.1}$$

so that $p+n = m$. Hence the redundancy of controls can be solved by defining a global state vector $\mathbf{z}$, obtained by appending the supplementary vector $\mathbf{c}$ (or $\mathbf{s}$) to the geometric vector $\mathbf{q}$, i.e. $\mathbf{z} = (\mathbf{q}|\mathbf{c})$(respectively $\mathbf{z} = (\mathbf{q}|\mathbf{s})$). Let us consider, as an example, the two systems represented in Fig. 3 where the same bi-phalangeal finger architecture is actuated by three (on the left) or four (on the right) muscles, respectively. While in

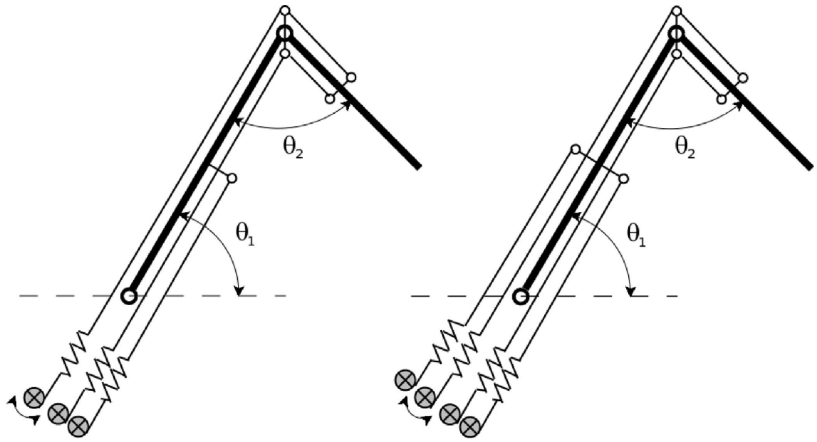

Figure 3.1: *Bi-phalangeal finger with two degrees of freedom actuated by three and four pseudomuscular actuators, respectively.*

the left case, only three controls are available, in order to complete the state vector, we can define only one generalized compliance

$$C = \left\| \frac{\partial \theta}{\partial P} \right\| \tag{3.2}$$

where θ is the vector (θ_1, θ_2) containing the two joint angles which characterize the state of the system and $P = (P_1, P_2)$ is the a bidirectional perturbation which can affect the system contained in its plane. In the right case four different controls can be used to define two different (and concentrated) compliances by choosing $y_1 : (\theta_1, \theta_2) \mapsto \theta_1$ and $y_2 : (\theta_1, \theta_2) \mapsto \theta_2$. The two canonical projections generate the two additional state variables

$$C_1 = \left\| \frac{\partial \theta_{\mathbf{1}}}{\partial P} \right\| \quad C_2 = \left\| \frac{\partial \theta_{\mathbf{2}}}{\partial P} \right\|. \tag{3.3}$$

In the static case, i.e. when the system is motionless and the geometrical state variables unchange, Eq. 3.1 can be easily computed by using the

Implicit Function Theorem. In fact, if

$$F(\mathbf{q}, P) = 0 \tag{3.4}$$

is the complete set of equations which describe the equilibrium of a system in its state $\mathbf{q}_0$ (i.e. a generalization of Eq. 3.8) with

$$det[\frac{\partial F}{\partial \mathbf{q}}(\mathbf{q}_0, P_0)] \neq 0$$

(trivially verified due to the consistence of the mechanical system), with $P_0 = 0$, by differentiating Eq. 3.4 we have:

$$\frac{\partial F}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial P} + \frac{\partial F}{\partial P} = 0$$

and

$$\frac{\partial \mathbf{q}}{\partial P} = -\left(\frac{\partial F}{\partial \mathbf{q}}\right)^{-1}\frac{\partial F}{\partial P}. \tag{3.5}$$

By pre-multiplying Eq. 3.5 by

$$\frac{\partial y_i(\mathbf{q})}{\partial \mathbf{q}} \quad i = 1 \dots p$$

and calculating the operator norms, we explicitly obtain the expressions for quantities 3.1.

$$C_i = \left\| -\frac{\partial y_i(\mathbf{q})}{\partial \mathbf{q}}\left(\frac{\partial F}{\partial \mathbf{q}}\right)^{-1}\frac{\partial F}{\partial P}\right\| \quad i = 1 \dots p \tag{3.6}$$

which represent all the compliances defined according to the $y_i(\mathbf{q})$ choice.

3.0.1 A Simple Example

A simple mechanical system implementing a monodimensional muscle model agonist-antagonist opposing an external force, is constituted by a motor linked two springs with quadratic characteristic (Fig. 3.2). These

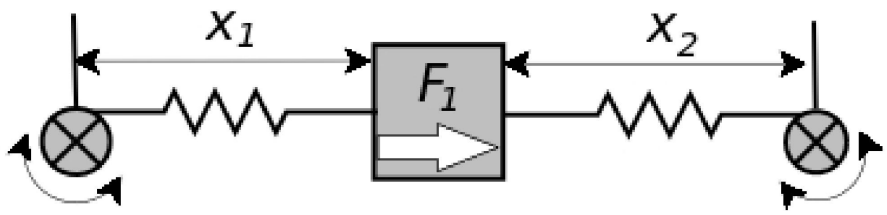

Figure 3.2: *A mechanical system with pseudo-muscular actuators.*

elements can be obtained, as an example, by using tapered compression springs having variable section, which work with packed wires. One end of each spring is connected to a load while the other one is linked to an inextensible cable pulled by the motor whose length cable is directly related to the parameter λ.

The equilibrium of the horizontal forces for the system shown in Fig. 3.2 is expressed by the following equation:

$$k_1(x_1 - \lambda_1)^2 - k_2(x_2 - \lambda_2)^2 - F_1 = 0 \tag{3.7}$$

where x_1 and x_2 are the distances between ends of the springs and the motors. If we assume that $x_1 + x_2 = l$ (constant length) and assign $x_1 = x$, Eq. 3.7 is clearly satisfied by infinitely many couples (λ_1, λ_2). But if we consider a perturbation P of the force which acts on the system, we have

$$k_1(x - \lambda_1)^2 - k_2(l - x - \lambda_2)^2 - F_1 + P = 0 \tag{3.8}$$

and if we define the following quantity:

$$S = \left| \frac{dP}{dx} \right| \tag{3.9}$$

the system constituted by Eqs. 3.7 and 3.9 with an assigned $S = S_0$, chosen according to the users requirement, is verified by a unique couple (λ_1, λ_2), for $x_1 > \lambda_1$ and $x_2 > \lambda_2$.

Important Observation: A non-linear characteristic of springs is required to obtain a hardware feed-forward control of the system compliance. In fact, it is easy to show that, if springs had linear characteristics, C value would not be a consequence of the choice of λs (the derivative of a linear expression is a constant value) and infinity couples (λ_1, λ_2) would produce the same x.

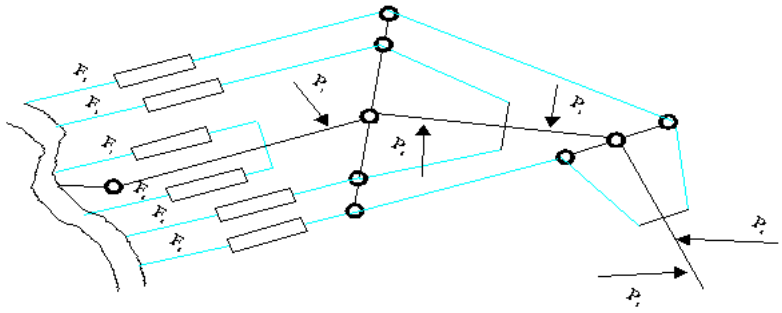

Figure 3.3: *Kinematic chain with n DOF actuated by a functional group.*

Chapter 4

The Dynamic Stiffness Operator

The possibility to co-activate muscles in order to change stiffness is a very useful mechanism used by a biological system to preserve himself from a certain perturbation, for example we can suppose that, either a biological system is strictly related with the environment where it operates and "knows" the possible common perturbation, or that it tries to protect itself against its own singularities, as perturbations characterized by the same natural frequencies of the system.

The acquisition and production of new informations on dynamical behaviour regarding the capability for muscles to handle external solicitations to adapt and preserve from perturbations can be persecuted via the definition of an enlarged concept of stiffness, which takes into account the dynamical terms of a kinematic chain during its motion. This new quantity, dynamically controlled by muscular co-activation, is introduced by the use of fundamental methods in functional analysis (e.g. Fréchet and Gâteaux differentiation) and allows a theoretical and practical ap-

proach to the study of the performances of a chain during collisions or under external perturbations.

4.1 Definition Of Dynamic Stiffness

The idea of stiffness for a kinematic chain in the dynamic case is substantially an extension of the corresponding one defined in the static case. When a kinematic chain is in motion, a load perturbation may produce changes in terms of trajectory and vice versa.

To define mathematically this concept, let's consider the following Hilbert vector spaces, provided by inner product:

$$\begin{aligned} X &= \{x(t)|x \text{ is a possible trajectory for the system}\} \\ P &= \{L(t)|L \text{ is a load applicable to the system}\} \end{aligned} \tag{4.1}$$

In this case, with usual notation, we indicate with T_{x_0} and T_{p_0}, if they exist, the tangent spaces to X in x_0 and P in p_0, respectively. T_{x_0} and T_{p_0} are Hilbert spaces with the induced norms.
Let's define two sets of Hilbert spaces very useful in the following:

$$T_X = \bigcup_{x_0 \in X} T_{x_0} \qquad T_P = \bigcup_{p_0 \in P} T_{p_0} \tag{4.2}$$

T_X and T_P represent the tangent bundles of spaces X and P respectively, in simple words, the sets of all tangent spaces in any point to X and P.

Then, given a control vector λ, it is possible to define the *Load map* L_λ which associates to each trajectory $x(t)$ the external load necessary to the system to perform it:

$$\begin{aligned} L_\lambda : \quad & X \longrightarrow P \\ & x(t) \mapsto L_{x,\lambda}(t) \end{aligned} \tag{4.3}$$

Definition: (Dynamic Stiffness Operator) *We define the Dynamic Stiffness Operator as the Fréchet differential (Gâteaux derivative in the case of dimension greater than one) of the map L_λ in respect to the trajectory $x(t)$.*

$$S_{x(t),\lambda} = \frac{\partial L_{x(t),\lambda}}{\partial x(t)} \tag{4.4}$$

The concepts of *Fréchet* and *Gâteaux differentiability* are defined in Appendix B.1, in simple words they extend the concept of differentiability of a function in respect to a variable to the more complex concept of differentiability of a functional in respect to a function (*functional derivative*).

In this way the Dynamic Stiffness Operator maps the tangent space $T_{x(t)}$ of the possible perturbations of trajectory $x(t) \in X$ into the tangent space $T_{L_{x,\lambda}(t)}$ of the possible perturbations of external load $L_{x,\lambda}(t) \in P$, necessary to perform $x(t)$ ($L_{x,\lambda}(t)$ is the image of $x(t)$ by the Load map L_λ):

$$\begin{aligned} S_{x(t),\lambda} : \quad & T_{x(t)} \longrightarrow T_{L_{x,\lambda}(t)} \\ & v(t) \mapsto S_{x(t),\lambda} \star v(t) \end{aligned} \tag{4.5}$$

Where the symbol $\star$ denotes a functional composition product, which meaning will be clear in the following computations.

In order to clarify the previous definitions and give a physical consistence to the concept of stiffness, let's take in consideration a kinematic

chain with one-dimensional trajectories, such as the system represented in Fig.3.2. This system can be described by the following equation:

$$\phi(x)\ddot{x} + \chi(x)\dot{x} + \psi(x) = G(x, t, \lambda) \tag{4.6}$$

where $x(t)$ is the status of the system at time t, $\phi(x), \chi(x), \psi(x)$ represent the ordinary coefficients of a mechanical second order system (i.e. the coefficients derived from the lagrangian of the system) and $G(x, t, \lambda)$ represents the generalized external forces applied to the system. In particular G can be expressed as:

$$G(x, \lambda) = H(x, \lambda) + K(t) \tag{4.7}$$

where $H(x, \lambda)$ represents the generalized forces exerted by muscles (controlled by λ) and $K(t)$ the external loads. Since $\phi(x)$ represents either a mass or an inertial momentum, we can suppose that $\phi(x) > 0$ and by dividing, it can be neglected in eq. 4.6.

$$\ddot{x} + \nu(x)\dot{x} + M(x, \lambda) = L_{x,\lambda}(t) \tag{4.8}$$

where: $\nu(x) = \chi(x)/\phi(x)$, $L_{x,\lambda}(t) = K(t)/\phi(x)$ and $M(x, \lambda) = -H(x, \lambda)/\phi(x) + \psi(x)/\phi(x)$.

The notation $L_{x,\lambda}(t)$ wants to emphasize the dependence of the function on the controls and on the trajectory realized $x(t)$.

Given $x(t)$ and $\lambda(t)$ we have a unique $L_{x,\lambda}(t)$ satisfying the equation. The map $x \mapsto L_{x,\lambda}$ between the space of trajectories and the space of loads, which are supposed time-Hilbert's spaces, is generally not linear and depends on the system geometry and on the muscle model.

The Fréchet differential of the Load map $L_{x,\lambda}$ acts from the space of all the perturbations to the trajectory $x(t)$ to the space of possible perturbations of loads necessary to perform them.

Taking into consideration the system expressed by eq. 4.6, let's calculate explicitly the stiffness operator, i.e. the Fréchet differential of the Load map, expressed in eq.4.8: let us consider a perturbation of trajectory by adding to $x(t)$ a small variation $\delta x(t)$. The load required to perform the new trajectory is given by:

$$\ddot{x} + \ddot{\delta x} + \nu(x + \delta x)\dot{x} + \nu(x + \delta x)\dot{\delta x}$$

$$+ M(x + \delta x, \lambda) = L_{x+\delta x,\lambda}(t). \tag{4.9}$$

By subtracting 4.8 and 4.9 and replacing the differences between corresponding terms by first order differentials, we obtain an explicit expression of the differential we were looking for:

$$\frac{\partial L_{x,\lambda}}{\partial x}[\delta x] = \ddot{\delta x} + \frac{\partial \nu(x)}{\partial x}\delta x\dot{x} + \nu(x)\dot{\delta x} + \frac{\partial M(x,\lambda)}{\partial x}\delta x \tag{4.10}$$

The operator $\frac{\partial L_{x,\lambda}}{\partial x}$ is the *Dynamic Stiffness Indicator* we consider (see Def.4.5).

Denoting $\delta x(t) = v(t)$ it is possible to express the composition function product $\star$ introduced in Def.4.5 in the following way:

$$\frac{\partial L_{x,\lambda}}{\partial x}[v(t)] = S_{x(t),\lambda} \star v(t) \tag{4.11}$$

Important Observation: *It is important to note that the Dynamic Stiffness Operator depends on the chosen controls λs. In particular, in kinematic chains actuated by using Feldman's muscles, different controls can perform the same trajectory even if stiffness or compliance result different.*

4.1.1 A Simple Example: Monodimensional Case

In order to provide an intuitive knowledge, let us consider again the example of dynamical system illustrated in Fig. 3.2.

Eq. 3.8 represents the equilibrium of the considered system when an external perturbation is applied. The corresponding relation when the system is in motion is given by:

$$\mu\ddot{x} + \nu\dot{x} + k_1(x - \lambda_1)^2 - k_2(l - x - \lambda_2)^2 = L(t) \tag{4.12}$$

where x is the trajectory performed by the chain, ν is the friction coefficient and $L(t)$ is a time function representing the external load applied and depending on controls and trajectory.

Applying the method described at the beginning of this section (*Gâteaux functional derivative*), we can calculate explicitly the stiffness operator for the system 4.12:

$$\mu\ddot{v} + \nu\dot{v} + 2k_1(x - \lambda_1)v + 2k_2(l - x - \lambda_2)v = S_{x,\lambda} \star v \tag{4.13}$$

where $v(t)$ is a generic perturbation of the trajectory $x(t) \in X$.

To estimate the stiffness for a generic perturbation $v(t)$, let us suppose to study the evolution of the system for 2π sec and suppose that $X = H^2(0, 2\pi)$ and $P = L^2(0, 2\pi)$ (Hilbert functional spaces coincide with all their tangent spaces so we can identify $T_{x(t)}$ with X and $T_{L_{x,\lambda}(t)}$ with P, see def.4.2). These spaces are endowed with the canonical scalar products (for definitions see Appendix A.3):

$$X \times X \ni (v, v) \mapsto < v, v >_X = \int_0^{2\pi} (v^2 + \dot{v}^2 + \ddot{v}^2)dt \in \mathbb{R} \tag{4.14}$$

$$P \times P \ni (f, f) \mapsto < f, f >_P = \int_0^{2\pi} f^2 dt \in \mathbb{R} \tag{4.15}$$

Scalar product will generally be denoted by $< \cdot, \cdot >$ when it does not generate ambiguities. It is easy to prove, by using trigonometric identities, that the following basis

$$\mathcal{B}_{H^2} = \left\{ c_0 = \tfrac{1}{\sqrt{2\pi}}, \quad s_k = \tfrac{\sin(kt)}{\sqrt{\pi(k^4+k^2+1)}}, \quad c_k = \tfrac{\cos(kt)}{\sqrt{\pi(k^4+k^2+1)}} \right\}_{k \in N^\star} \tag{4.16}$$

$$\mathcal{B}_{L^2} = \left\{ c_0 = \frac{1}{\sqrt{2\pi}}, s_k = \frac{\sin(kt)}{\sqrt{\pi}}, c_k = \frac{\cos(kt)}{\sqrt{\pi}} \right\}_{k \in N^\star} \tag{4.17}$$

are two ortho-normal complete systems for $H^2(0, 2\pi)$ and $L^2(0, 2\pi)$, respectively, so every element in X and P can be approximated by a (finite) linear combination of their elements (see Appendix A.2). For simplicity of notations we have set, for any non negative integer k

$$\begin{cases} c_0^{H^2} = \frac{1}{\sqrt{2\pi}} \\ s_k^{H^2} = \frac{\sin(kt)}{\sqrt{\pi(k^4+k^2+1)}} \\ c_k^{H^2} = \frac{\cos(kt)}{\sqrt{\pi(k^4+k^2+1)}} \end{cases} \tag{4.18}$$

and

$$\begin{cases} c_0^{L^2} = \frac{1}{\sqrt{2\pi}} \\ s_k^{L^2} = \frac{\sin(kt)}{\sqrt{\pi}} \\ c_k^{L^2} = \frac{\cos(kt)}{\sqrt{\pi}} \end{cases} \tag{4.19}$$

Computer Calculations

Numerical simulations have been executed with MAPLE (see Appendix E.1 for code). We will show the following two examples:

1. For different couples of controls, the Dynamic Stiffness Operator assumes different values, even if the trajectory performed is the same.

2. Given piecewise constant controls, the stiffness of the system changes with respect of the frequency of the perturbation.

Case 1

Let us consider the system of Fig. 3.2, having set $l = 10\,m$, $F_1 = 4\,N$, $\mu = 1\,kg$, $\nu = 1\,N/m$, $k_1 = k_2 = 1\,N/m^2$ and let us suppose that system 4.12 performs the trajectory $x(t) = 5\,m$.

Let us consider the following couples of controls (expressed in m). $\lambda(t)$ given by:

$$\begin{cases} \lambda_1 = 3 - t \\ \lambda_2 = 5 + \sqrt{4t + t^2} \end{cases} \tag{4.20}$$

and $\lambda'(t)$ given by:

$$\begin{cases} \lambda'_1 = 1 - 2t \\ \lambda'_2 = 5 + 2\sqrt{3 + 4t + t^2} \end{cases} \tag{4.21}$$

To estimate the stiffness for 2π seconds, let's calculate the images in P of $c_k^{H^2}$ and $s_k^{H^2} \in X$ (def.4.18 and 4.19) by $S_{\lambda,x}$ and $S_{\lambda',x}$. The stiffness for the first couple of controls is:

$$\begin{cases} S_{x,\lambda} \star s_k^{H^2} = \frac{s_k^{L^2}\left(2t+4-k^2-2\sqrt{4t+t^2}\right)}{sqrt1+k^2+k^4} \\ S_{x,\lambda} \star c_k^{H^2} = \frac{c_k^{L^2}\left(2t+4-k^2-2\sqrt{4t+t^2}\right)}{sqrt1+k^2+k^4} \end{cases} \tag{4.22}$$

and for the second couple of controls is:

$$\begin{cases} S_{x,\lambda'} \star s_k^{H^2} = \frac{s_k^{L^2}\left(4t+8-k^2-2\sqrt{3+4t+t^2}\right)}{sqrt1+k^2+k^4} \\ S_{x,\lambda'} \star c_k^{H^2} = \frac{c_k^{L^2}\left(4t+8-k^2-2\sqrt{3+4t+t^2}\right)}{sqrt1+k^2+k^4} \end{cases} \tag{4.23}$$

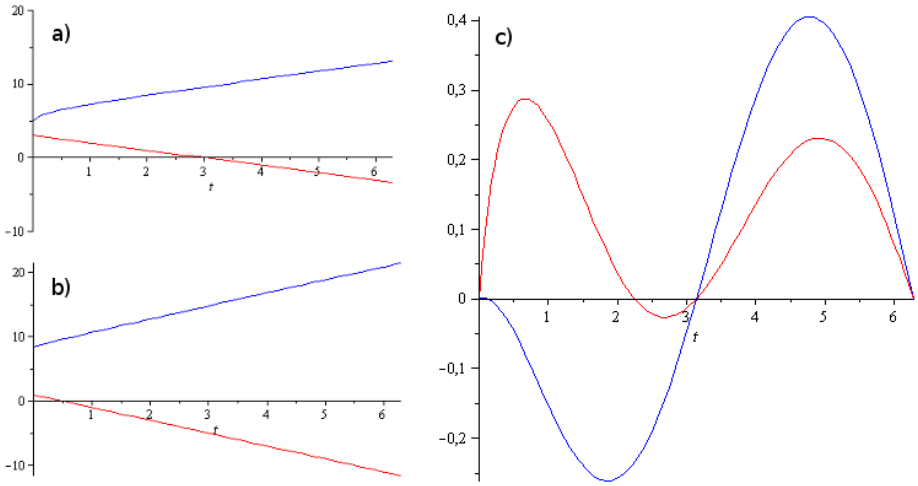

Figure 4.1: *a) Trajectories of controls $\lambda_1(t)$ (red line) and $\lambda_2(t)$ (blue line). b) Trajectories of controls $\lambda_1'(t)$ (red line) and $\lambda_2'(t)$ (blue line). c) Dynamical Stiffness of the system in presence of the trajectory perturbation $v(t) = \sin(t)$ with the couples of controls $\lambda(t)$ (red line) and $\lambda'(t)$ (blue line).*

We can observe that, for different couples of controls, the Dynamic Stiffness Operator assumes different values for all the vectors of the basis A.2, and recalling that this basis is a ortho-normal complete systems (see Appendix A.2), the dynamic stiffness is different in the whole space X, even if the trajectory performed is the same.

In Fig.4.1 are represented the trajectories of the two couples of controls and the respective values of mechanical stiffness of the system in presence of the trajectory perturbation $v(t) = \sin(t)$.

Case 2

If controls λs are chosen among piecewise constant functions, eq.4.13 reduces to a linear second order differential equation solvable by the clas-

sical linear system theory. In particular, having set the same values of previous calculation: $l = 10\ m$, $F_1 = 4\ N$, $\mu = 1\ kg$, $\nu = 1\ N/m$, $k_1 = k_2 = 1\ N/m^2$, let us suppose that system 4.12 is in equilibrium in $x = 5\ m$ for $t = 0$. For $t < 2$ we have set controls $\lambda_1 = 3\ m$ and $\lambda_2 = 5\ m$; at $t = 2$ controls assume values $\lambda_1 = 5\ m$ and $\lambda_2 = 6\ m$ definitively. According to the assignment values, system 4.12 becomes:

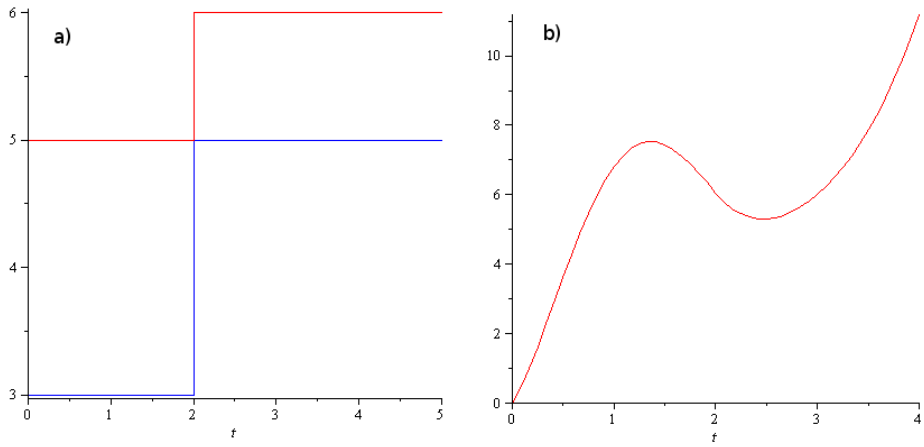

Figure 4.2: *a) Controls $\lambda_1(t)$ (blue line) and $\lambda_2(t)$ (red line). b)Trajectory performed by the system.*

$$\begin{cases} \ddot{x}(t) + \dot{x}(t) + (x(t) - \lambda_1(t))^2 - (10 - x(t) - \lambda_2(t))^2 = 4 \\ x(0) = 5 \\ \dot{x}(0) = 0 \end{cases} \tag{4.24}$$

and it evolves according to the following continuous trajectory (fig.4.2):

$$x(t)=\begin{cases} \frac{1}{3}e^{-\frac{t}{2}}\sin\left(\frac{t\sqrt{15}}{2}\right)\sqrt{15}-5e^{-\frac{t}{2}}\cos\left(\frac{t\sqrt{15}}{2}\right)+5 & t<2 \\ \\ -\frac{1}{3}e^{3-2t}\left(10\cos(\sqrt{15})+2\sqrt{15}\sin(\sqrt{15})\right) & \\ -\frac{1}{3}e^{t-3}\left(5\cos(\sqrt{15})+3\sqrt{15}\sin(\sqrt{15})\right) & t\geq 2 \\ +\frac{5}{3}e^{-2+t}+\frac{5}{6}e^{4-2t}+\frac{5}{2} & \end{cases} \tag{4.25}$$

The stiffness $S_{\lambda,x}$ is univocalluy determined by the images in P of $c_k^{H^2}$ and $s_k^{H^2}$ (see def.4.18 and 4.19).

$$\frac{\partial L_{x,\lambda}}{\partial x}[c_k^{H^2}]=\begin{cases} \frac{-c_k^{L^2}k^2-s_k^{L^2}k^2+4c_k^{L^2}k^2}{k^4+k^2+1} & t<2 \\ \frac{-c_k^{L^2}k^2-s_k^{L^2}k^2-2c_k^{L^2}k^2}{k^4+k^2+1} & t\geq 2 \end{cases} \tag{4.26}$$

and

$$\frac{\partial L_{x,\lambda}}{\partial x}[s_k^{H^2}]=\begin{cases} \frac{-s_k^{L^2}k^2+c_k^{L^2}k^2+4s_k^{L^2}k^2}{k^4+k^2+1} & t<2 \\ \frac{-s_k^{L^2}k^2+c_k^{L^2}k^2-2s_k^{L^2}k^2}{k^4+k^2+1} & t\geq 2 \end{cases} \tag{4.27}$$

We can observe that the images of the Stiffness Operator change in respect to the frequency of perturbation of the trajectory. In Fig.4.3 are plotted values of stiffness in presence of perturbations with increasing frequency.

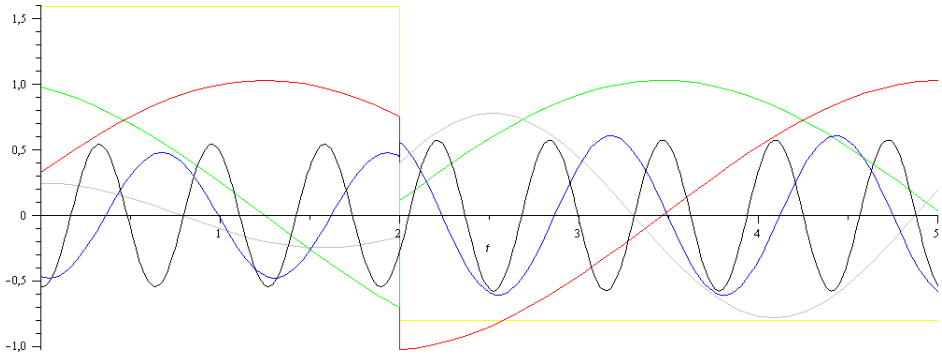

Figure 4.3: *Dynamical Stiffness of the system with trajectory perturbations* $c_0^{H^2}$ *(yellow line),* $s_1^{H^2}$ *(red line),* $c_1^{H^2}$ *(green line),* $s_2^{H^2}$ *(grey line),* $c_5^{H^2}$ *(blue line) and* $c_{10}^{H^2}$ *(black line).*

Chapter 5

Evaluation of Dynamic Stiffness

The dynamic Stiffness Operator can be useful to evaluate the stiffness of a kinematic chain in motion in a continue way, depending on the trajectory and on the external forces applied.

5.1 Bidimensional Case: Biphalangeal Finger

Let's consider the problem of the position and compliance control of a manipulator with two arms, two joints and three actuators, representing a biphalangeal finger performing planar trajectories.

The system is represented in the figures 5.1 and 5.2. The points O_1 and O_2 represent the centers of the metacarpal-phalangeal and proximal-interphalangeal joints; a_1 and a_2 are the lengths of the proximal and middle phalanx respectively and θ_1 and θ_2 represent the two angles of the joints of the manipulator (as illustrated in fig. 5.2). F_1 represents the

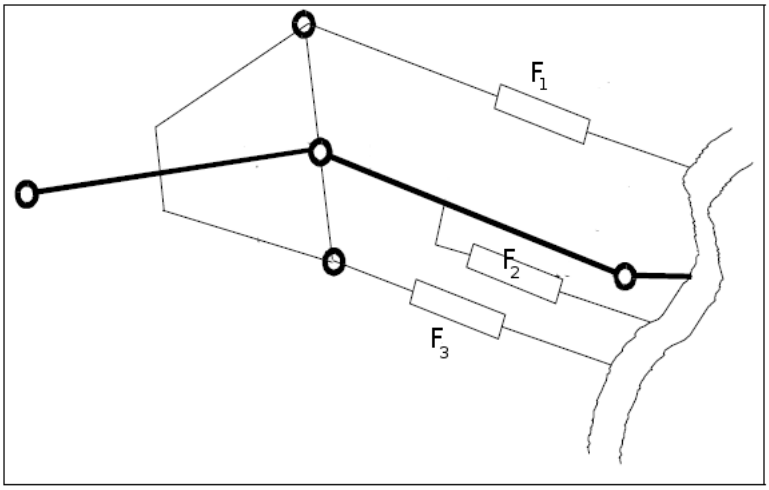

Figure 5.1: *Bi-phalangeal finger with two degrees of freedom and actuated by three actuators.*

force exerted by the extensor muscle, F_2 and F_3 are the forces exerted by the two flessors: they are parallel to the axis of the phalanxes on which respective tendons insert.

For a system in an equilibrium point the *Cardinal Equations* hold:

$$\begin{aligned} \sum F &= 0 \\ \sum M_O &= 0 \end{aligned} \tag{5.1}$$

where F and M_O represent respectively the external forces and the moments of the external forces relative to a fixed pole O.

To obtain the equations of the system in motion we consider the *Lagrange Equations*:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \xi_i \qquad i = 1, 2 \tag{5.2}$$

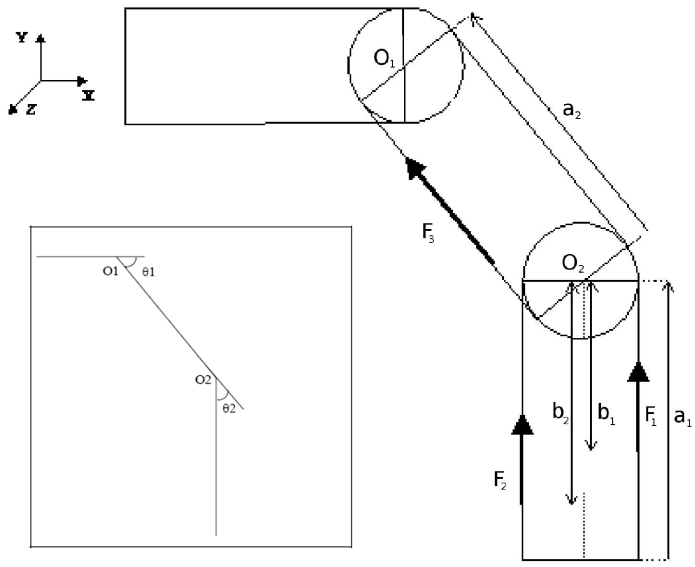

Figure 5.2: *Parametrization of planar trajectories of the manipulator.*

where $L = T - U$ is called *Lagrangian* of the system (T =kinetic energy, U =potential energy), q_i are the generalized coordinates of the system (we can consider $q_i = \theta_i$ where $i = 1, 2$) and ξ_i represent the generalized forces acting on the joint i.

The *Kinetic Energy* of the chain is the sum of the translational and rotational contributions:

$$T_i = \frac{1}{2} m_i \nu_{Gi}^T \nu_{Gi} + \frac{1}{2} \omega_1^T R_i I_{li}^i R_i^T \omega_i \qquad i = 1, 2 \tag{5.3}$$

where m_i is the mass of the arm i, ν_{Gi}^T is the linear velocity of the baricentrum of the arm i, ω_i is the angular velocity of the arm i, I_{li}^i is the inertial tensor relative to the arm i expressed in the coordinate system associated to the arm and R are the matrices of rotation.

The *Potential Energy* of the chain is expressed as follows:

$$U_i = -m_i g_0^T p_{li} \qquad i = 1, 2 \tag{5.4}$$

where g_0 is the gravity vector in the inertial reference and p_{li} represent the baricentrum position vector of the arm i.

So we have two lagrangian equations (eq. 5.2) one for each joint, which can be expressed in the explicit form as follows (for better explanations see [37]):

$$\begin{gathered}[I_{l1} + m_1 l_1^2 + I_{l2} + m_2(a_1^2 + l_2^2 + 2a_1 l_2 \cos(\theta_2))]\ddot{\theta}_1 + \\ +[I_{l2} + m_2(l_2^2 + a_1 l_2 \cos(\theta_2))]\ddot{\theta}_2 - 2m_2 a_1 l_2 \sin(\theta_2)\dot{\theta}_1\dot{\theta}_2 + \\ -m_2 a_1 l_2 \sin(\theta_2)\dot{\theta}_2^{\,2} + (m_1 l_1 + m_2 a_1) g \cos(\theta_1) + \\ +m_2 l_2 g \cos(\theta_1 + \theta_2) = \xi_1\end{gathered} \tag{5.5}$$

$$\begin{gathered}[I_{l2} + m_2(l_2^2 + a_1 l_2 \cos(\theta_2))]\ddot{\theta}_1 + (I_{l2} + m_2 l_2)\ddot{\theta}_2 + \\ +m_2 a_1 l_2 \sin(\theta_2)\dot{\theta}_1^{\,2} + m_2 l_2 g \cos(\theta_1 + \theta_2) = \xi_2\end{gathered} \tag{5.6}$$

where

$$\xi_1(\theta_1, \theta_2) = (b_{F1}^{(1)} \times F1 + b_{F2}^{(1)} \times F2 + b_{F3}^{(1)} \times F3 + b_C^{(1)} \times C) \tag{5.7}$$

$$\xi_2(\theta_1, \theta_2) = (b_{F1}^{(2)} \times F1 + b_{F2}^{(2)} \times F2 + b_{F3}^{(2)} \times F3 + b_C^{(2)} \times C) \tag{5.8}$$

Fi Force exerted by the actuator i ($i = 1, 2$) (fig.5.1).

$b_{Fi}^{(j)}$ Arm of the force exerted by the actuator i in respect to the pole j ($i, j = 1, 2$).

C External load applied to the manipulator (at the extremity).

bC_j Arm of the external load applied to the manipulator in respect to the pole j ($i, j = 1, 2$).

The Dynamic Stiffness Operator is represented by the functional derivative of the external load applied to the chain (eq. 4.13).

Let's consider the following matrix:

$$\begin{bmatrix} \dfrac{\partial C_x}{\partial \theta_1(t)} & \dfrac{\partial C_x}{\partial \theta_2(t)} \\ \dfrac{\partial C_y}{\partial \theta_1(t)} & \dfrac{\partial C_y}{\partial \theta_2(t)} \end{bmatrix}$$

The Stiffness Operator applied to a vector $(v_i(t), v_2(t))$ (representing a generic perturbation of the two-dimensional trajectory of the manipulator) is a bidimensional vector of functions, given by the composition between functional matrices, defined as follows:

$$\begin{bmatrix} \frac{\partial C_x}{\partial \theta_1} & \frac{\partial C_x}{\partial \theta_2} \\ \frac{\partial C_y}{\partial \theta_1} & \frac{\partial C_y}{\partial \theta_2} \end{bmatrix} \star \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \stackrel{def}{=} \begin{bmatrix} \frac{\partial C_x}{\partial \theta_1} \star v_1(t) + \frac{\partial C_x}{\partial \theta_2} \star v_2(t) \\ \frac{\partial C_y}{\partial \theta_1} \star v_1(t) + \frac{\partial C_y}{\partial \theta_2} \star v_2(t) \end{bmatrix} \tag{5.9}$$

$\frac{\partial C_i}{\partial \theta_j}$ $(i = x, y \ j = 1, 2)$ represents the partial functional derivative (Appendix B.1) of the horizontal (or vertical) component of the external load in respect to the trajectory of the joint j.

$v_i(t)$ $(i = 1, 2)$ is an element of the space of functions $T(X)$ and represents a generic perturbation of the trajectory $\theta_i(t)$.

So, in the bidimensional case, for every perturbation of the bidimensional trajectory of the manipulator, we obtain a bidimensional vector of functions representing the Stiffness Operator. The two components depend on controls and on the trajectory of the system:

$$\begin{aligned} S^x_{\lambda,\theta} \star v(t) &= \frac{\partial C_x}{\partial \theta_1} \star v_1(t) + \frac{\partial C_x}{\partial \theta_2} \star v_2(t) \\ S^y_{\lambda,\theta} \star v(t) &= \frac{\partial C_y}{\partial \theta_1} \star v_1(t) + \frac{\partial C_y}{\partial \theta_2} \star v_2(t) \end{aligned} \tag{5.10}$$

5.1.1 Computer Calculations

In respect to the bidimensional case we have done many numerical simulations with MAPLE (see Appendix E.2 for code).

Let us consider the system with bidimensional trajectory represented in Fig. 5.1; the phalanges are represented by cylinders, and are actuated by three polymeric actuators following the Feldman's quadratic law. The equations of motion (eq.5.5, eq.5.6) implemented in MAPLE are the following:

$$
\begin{aligned}
&> \mathbf{eq1} := \mathbf{eval}(\mathbf{expand}((\mathbf{I1} + \mathbf{m1} * \mathbf{l1^2} + \mathbf{m2} * ((\mathbf{l1/2})^2 \\
&+\mathbf{l2^2} + \mathbf{2} * \mathbf{l2} * \mathbf{l1/2} * \mathbf{cos}(\mathbf{theta2}))) * \mathbf{diff}(\mathbf{diff}(\mathbf{theta1}, \mathbf{t}), \mathbf{t}) \\
&+(\mathbf{I2} + \mathbf{m2} * (\mathbf{l2^2} + \mathbf{l2} * \mathbf{l1/2} * \mathbf{cos}(\mathbf{theta2})))* \\
&\mathbf{diff}(\mathbf{diff}(\mathbf{theta2}, \mathbf{t}), \mathbf{t}) - \mathbf{2} * \mathbf{m2} * \mathbf{l2} * \mathbf{l1/2} * \mathbf{sin}(\mathbf{theta2})* \\
&\mathbf{diff}(\mathbf{theta1}, \mathbf{t}) * \mathbf{diff}(\mathbf{theta2}, \mathbf{t}) - \mathbf{m2} * \mathbf{l2} * \mathbf{l1/2} * \mathbf{sin}(\mathbf{theta2})* \\
&\mathbf{diff}(\mathbf{theta2}, \mathbf{t})^2 + (\mathbf{m1} * \mathbf{l1} + \mathbf{m2} * \mathbf{l1/2}) * \mathbf{g} * \mathbf{cos}(\mathbf{theta1}) \\
&+\mathbf{m2} * \mathbf{l2} * \mathbf{g} * \mathbf{cos}(\mathbf{theta1} + \mathbf{theta2}) - \mathbf{xi1}, \mathbf{trig})) : \\
\\
&> \mathbf{eq2} := \mathbf{eval}((\mathbf{I2} + \mathbf{m2} * (\mathbf{l2^2} + \mathbf{l2} * \mathbf{l1/2} * \mathbf{cos}(\mathbf{theta2})))* \\
&\mathbf{diff}(\mathbf{diff}(\mathbf{theta1}, \mathbf{t}), \mathbf{t}) + (\mathbf{I2} + \mathbf{m2} * \mathbf{l2}) * \mathbf{diff}(\mathbf{diff}(\mathbf{theta2}, \mathbf{t}), \mathbf{t}) \\
&+\mathbf{m2} * \mathbf{l2} * \mathbf{l1/2} * \mathbf{sin}(\mathbf{theta2}) * \mathbf{diff}(\mathbf{theta1}, \mathbf{t})^2 \\
&+\mathbf{m2} * \mathbf{l2} * \mathbf{g} * \mathbf{cos}(\mathbf{theta1} + \mathbf{theta2}) - \mathbf{xi2}) :
\end{aligned}
\tag{5.11}
$$

The program solves the system in respect to the external loads applied to the extremity of the chain: C_x and C_y (component of the vector C).

$$
\begin{aligned}
&> \mathbf{sol} := \mathbf{solve}(\mathbf{eq1}, \mathbf{eq2}, \mathbf{Cx}, \mathbf{Cy}) : \\
&> \mathbf{CY} := \mathbf{rhs}(\mathbf{sol}[\mathbf{1}]) : \mathbf{CX} := \mathbf{rhs}(\mathbf{sol}[\mathbf{2}]) :
\end{aligned}
\tag{5.12}
$$

And then we can derive the expressions obtained by Gâteaux Differential (Appendix B.1). At the beginning we evaluate the two *difference quotients* for each component of C in respect of the variations of trajec-

tory of the first and second joint. The *difference quotients* are obtained by substituting to the trajectory $\theta = (\theta_1, \theta_2)$, the trajectory perturbed $\theta + \delta\theta = (\theta_1 + h_1 v_1, \theta_1 + h_1 v_1)$, where $v = (v_1, v_2)$ is a vector of the space (X def.4.2) and (h_1, h_2) is a couple of real numbers, and subtracting the new expression to the original one, and then dividing the differences by the modules of the increments of trajectory h_1 and h_2. We have already applied the same method in the monodimensional case (eq. 4.8, eq. 4.9), the difference now is that we have to evaluate two *difference quotients* for C_x and two for C_y, respectively related to increment $(h_1 v_1)$ of the trajectory θ_1 of the first joint and to increment $(h_2 v_2)$ of the trajectory θ_2 of the second joint.

$$
\begin{aligned}
&> \quad \mathbf{CXsj} := \mathbf{subs}(\mathbf{thetaj} = \mathbf{thetaj} + \mathbf{hj} * \mathbf{vj}(\mathbf{t}), \mathbf{CX}) : \quad j = 1, 2 \\
&> \quad \mathbf{CYsj} := \mathbf{subs}(\mathbf{thetaj} = \mathbf{thetaj} + \mathbf{hj} * \mathbf{vj}(\mathbf{t}), \mathbf{CY}) : \\
&> \quad \mathbf{rapincij} := \mathbf{simplify}((\mathbf{CXsj} - \mathbf{CX})/\mathbf{hj}) : \qquad i = 1 \quad j = 1, 2 \\
&> \quad \mathbf{rapincij} := \mathbf{simplify}((\mathbf{CYsj} - \mathbf{CY})/\mathbf{hj}) : \qquad i = 2 \quad j = 1, 2
\end{aligned}
\tag{5.13}
$$

To obtain the Gâteaux Differentials we have to evaluate the limits of the *difference quotients* when h_1 and h_2 tend to 0.

$$
>\mathbf{DiffGatij} := \mathbf{limit}(\mathbf{rapincij}, \mathbf{hj} = \mathbf{0}) : \qquad i = 1, 2 \quad j = 1, 2 \tag{5.14}
$$

Finally, the two components of the Dynamical Stiffness Operator (def. 5.10) are given by the sum of the Gâteaux Differentials of the horizontal and the vertical components of the external load:

$$
\begin{aligned}
&> \mathbf{Stiffx} := \mathbf{DiffGat11} + \mathbf{DiffGat12} : \\
&> \mathbf{Stiffy} := \mathbf{DiffGat21} + \mathbf{DiffGat22} :
\end{aligned}
\tag{5.15}
$$

In this way, given a trajectory perturbation, it is possible to determine the external load perturbation applied to the extremity of the chain which causes it.

Evaluation of Stiffness

Let's consider the system in Fig.5.1 and set parameters as follows: $l_1 = 2\,m$, $l_2 = 1\,m$ are the lengths of the first and the second phalange respectively, the rays are the same $r = 0.1\,m$, the masses are $m_1 = 2\,kg$, $m_2 = 1\,kg$, for simplicity we can assume the absence of attrite $\nu_1 = \nu_2 = 0\,N/m$, and the elastic constants of actuators are $k_1 = 2N/m^2$, $k_2 = k_3 = 1N/m^2$. We can assume as external load, applied at the extremity of the chain, the vector $C = (0, -20)$ (in Newtons). We can verify that, choosing a particular trajectory, the two components of the Dynamical Stiffness of the system vary in respect to the controls applied.

Let us consider the following triplets of controls. $\lambda(t)$ given by:

$$\begin{cases} \lambda_1(t) = 0,09 \cdot t - 0,87 \\ \lambda_2(t) = 1 - \frac{t}{2} \\ \lambda_3(t) = 1 + \frac{1}{2}\sqrt{33,4 + 3,63 \cdot t - 0,8 \cdot t^2} \end{cases} \tag{5.16}$$

and $\lambda'(t)$ given by:

$$\begin{cases} \lambda'_1(t) = 0,65 \cdot t - 1,13 \\ \lambda'_2(t) = 5 - t \\ \lambda'_3(t) = 1 + \frac{1}{2}\sqrt{4,8 + 2,5 \cdot t - 0,34 \cdot t^2} \end{cases} \tag{5.17}$$

for simplicity of notations we assume

$$\omega = \sqrt{33,4 + 3,63 \cdot t - 0,8 \cdot t^2} \qquad \omega' = \sqrt{4,8 + 2,5 \cdot t - 0,34 \cdot t^2}$$

Obviously we have to take in consideration the existence conditions of square roots, which are illustrated in Fig.5.3 (for time $t \in [0, 8]$ we don't have any problem in respect to the existence of roots).

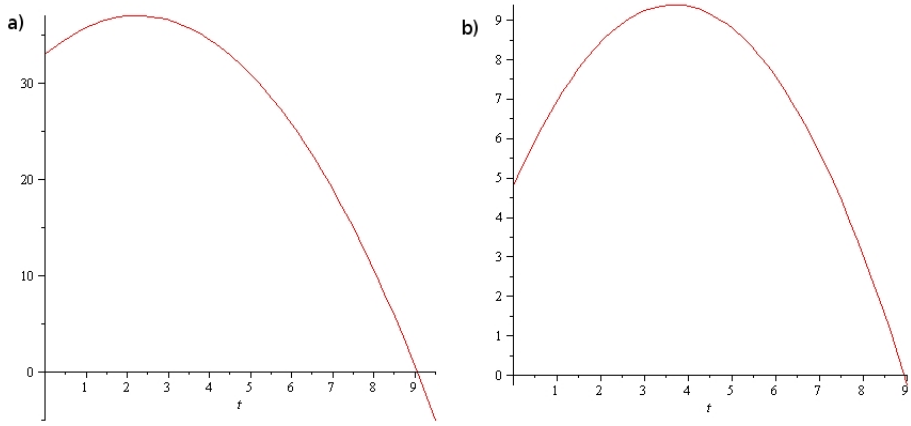

Figure 5.3: *a) Argument of square root of control $\lambda_3(t)$. b) Argument of square root of control $\lambda_3'(t)$.*

We assume that $\theta(t) = (\theta_1(t), \theta_2(t)) = (0, \pi/3)$ is the bidimensional trajectory of the system.

The two Dynamical Stiffness Operators $Stiff_\lambda$ and $Stiff_{\lambda'}$ associated to controls λ and λ' respectively (see def.5.15) are very complex expressions. We can observe that the images of a generic trajectory perturbation $v(t)$ by the Stiffness Operators are different functions (Fig.5.4), representing the load variations necessary to obtain the perturbation $v(t)$ with controls $\lambda(t)$ and $\lambda'(t)$.

Let us consider the trajectory perturbation $v(t) = (\sin(t), \cos(t))$. The images by Stiffness Operators relative to controls $\lambda(t)$ and $\lambda'(t)$ are given by:

$$Stiff_\lambda \star v(t) = \begin{bmatrix} 0,61\sin(t)\omega - 0,87 \cdot t^2\cos(t) - 1,65 \cdot t^2\sin(t) + \\ +7,58 \cdot t\sin(t) + 4 \cdot t\cos(t) - 183,9\sin(t) - 109,72\cos(t) \\ \\ (\sin(t) + \cos(t))\left(0,61\omega + 27,8 + 11 \cdot t + 21,2 \cdot t^2\right) \end{bmatrix}$$

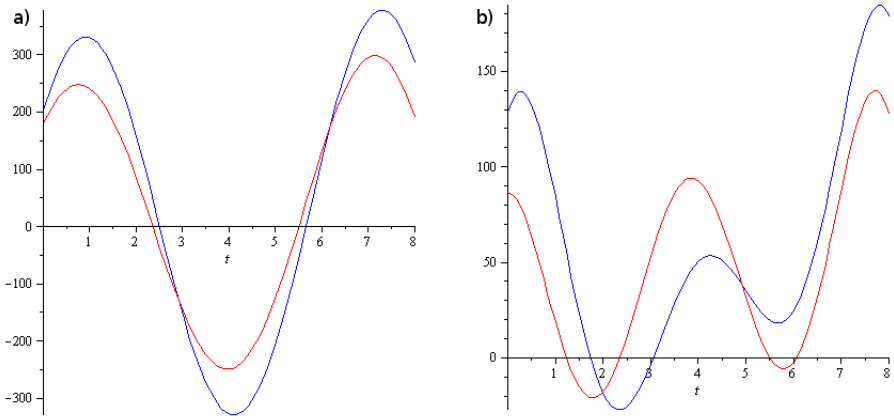

Figure 5.4: *Dynamical Stiffness of the biphalangeal finger ($Stiff_x$ =blue line, $Stiff_y$=red line) relative to the trajectory perturbation $v(t) = (\sin(t), \cos(t))$ a)with controls $\lambda(t)$ b)with controls $\lambda'(t)$.*

$$Stiff_{\lambda'} \star v(t) = \begin{bmatrix} 2,45\sin(t)\omega' + 11,4 \cdot t^2 \sin(t) - 43,8 \cdot t\cos(t) + \\ -82,9 \cdot t\sin(t) + 5,9 \cdot t^2\cos(t) + 91,5\sin(t) + 61\cos(t) \\ \\ (\sin(t) + \cos(t))\left(2,45\omega' + 159,84 + 110,6 \cdot t + 145,52 \cdot t^2\right) \end{bmatrix}$$

5.2 Threedimensional Case: Anthropomorphic Manipulator

An Anthropomorphic Manipulator is represented by a kinematic chain with three arms and three (rotoidal) joints (see Fig.5.5).

It's dynamical equation can be expressed by the following formula [37], which represents the *dynamical model of a manipulator in the joint's space*:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F_v}\dot{\mathbf{q}} + \mathbf{f_s}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \tau + \mathbf{J^T}(\mathbf{q})\mathbf{h_e} \tag{5.18}$$

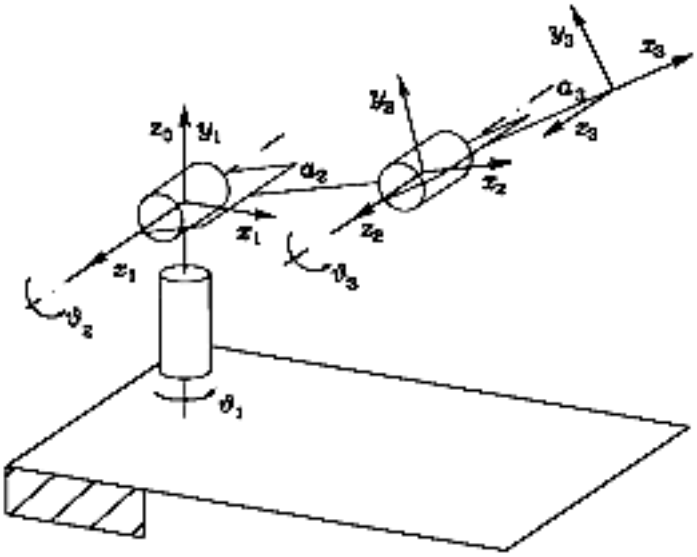

Figure 5.5: *Representation of the Anthropomorphic Manipulator.*

q Three-dimensional vector of the joint variables $\mathbf{q} = (q_1, q_2, q_3)$.

$B(q)$ Inertial matrix (3×3).

$C(d, \dot{q})$ Matrix of Centrifugal force and Coriolis effect (3×3).

F_v Diagonal matrix (3×3) of coefficients of viscous friction.

$f_s(\dot{q})$ $\mathbf{f_s} = \mathbf{F_s}\mathbf{sgn}(\dot{\mathbf{q}})$ where $\mathbf{F_s}$ is the diagonal matrix (3×3) of coefficients of static friction.

g Three-dimensional vector representing the gravitational force.

τ Three-dimensional vector representing the actuation forces (forces exerted by actuators).

$J^T(q)$ Jacobian Matrix (6×3).

h_e Six-dimensional vector representing the external forces and the external moments applied to the extremity of the manipulator $\mathbf{h_e} = (\zeta_x, \zeta_y, \zeta_z, \rho, \theta, \phi)$.

The manipulator has three rotoidal joints, which can be actuated by three different actuators. We can use actuators with a couple of quadratic springs (see Fig.5.6), controllable in stiffness.

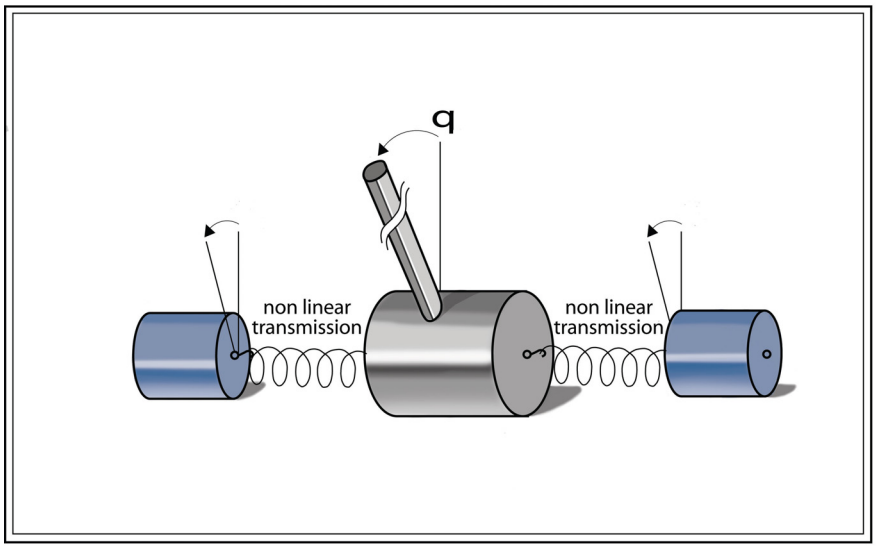

Figure 5.6: *Actuaror controlled by a couple of non-linear springs.*

The equations of the forces exerted by actuators can be expressed by the following formula:

$$\tau_i = k_{2i-1}(q_i - \lambda_{2i-1})^2 - k_{2i}(q_i - \lambda_{2i})^2 \qquad i = 1..3 \tag{5.19}$$

Where k_j and λ_j $(j = 1..6)$ are respectively the elastic constants of the six springs (two for every actuator) and the rest angles of the springs.

The system has rank three so we can control only three variables. The vector $\mathbf{h_e}$, representing the external forces and moments acting on the extremity of the chain, has instead dimension six, so we have to choose which components of $\mathbf{h_e}$ we would like to control.

As an example, we could the position of the end-effector of the kinematic chain in the Euclidean space $(\zeta_x, \zeta_y, \zeta_z)$, don't taking care on its rotations.

The dynamical controls of the system are six (the λs), the equation in the joint space are three (Eq.5.18, one for every dimension), so to complete the system we need three other equations: the equations representing the stiffness of the system, relative to the three external load perturbation components in the euclidean space.

The external forces applied to the end-effector are:

$$h_e = \begin{bmatrix} \zeta_x(q_1, q_2, q_3) \\ \zeta_y(q_1, q_2, q_3) \\ \zeta_z(q_1, q_2, q_3) \end{bmatrix}$$

The Dynamical Stiffness Operator is represented by the functional derivative of the components ζ_i $(i = 1..3)$ of the external forced applied to the chain.

Let's consider the following matrix (of partial derivatives):

$$\begin{bmatrix} \frac{\partial \zeta_x}{\partial q_1(t)} & \frac{\partial \zeta_x}{\partial q_2(t)} & \frac{\partial \zeta_x}{\partial q_3(t)} \\ \frac{\partial \zeta_y}{\partial q_1(t)} & \frac{\partial \zeta_y}{\partial q_2(t)} & \frac{\partial \zeta_y}{\partial q_3(t)} \\ \frac{\partial \zeta_z}{\partial q_1(t)} & \frac{\partial \zeta_z}{\partial q_2(t)} & \frac{\partial \zeta_z}{\partial q_3(t)} \end{bmatrix}$$

The Stiffness Operator applied to a generic trajectory perturbation $(v_1(t), v_2(t), v_3(t))$ is a three-dimensional vector of functions, defined as follows:

$$\begin{bmatrix} \frac{\partial \zeta_x}{\partial q_1} & \frac{\partial \zeta_x}{\partial q_2} & \frac{\partial \zeta_x}{\partial q_3} \\ \frac{\partial \zeta_y}{\partial q_1} & \frac{\partial \zeta_y}{\partial q_2} & \frac{\partial \zeta_y}{\partial q_3} \\ \frac{\partial \zeta_z}{\partial q_1} & \frac{\partial \zeta_z}{\partial q_2} & \frac{\partial \zeta_z}{\partial q_3} \end{bmatrix} \star \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(3) \end{bmatrix} \stackrel{def}{=} \begin{bmatrix} \frac{\partial \zeta_x}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_x}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_x}{\partial q_3} \star v_3(t) \\ \frac{\partial \zeta_y}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_y}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_y}{\partial q_3} \star v_3(t) \\ \frac{\partial \zeta_z}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_z}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_z}{\partial q_3} \star v_3(t) \end{bmatrix} \tag{5.20}$$

$\frac{\partial \zeta_i}{\partial q_j}$ $(i = x, y, z \;\; j = 1, 2, 3)$ represents the partial functional derivative (Appendix B.1) of the i-th component of the external load applied to the manipulator in respect to the trajectory of the joint j.

$v_i(t)$ $(i = 1, 2, 3)$ represents a generic perturbation of the trajectory q_i of the i-th joint.

For an Anthropomorphic Manipulator, in respect to every perturbation of its three-dimensional trajectory, the Stiffness Operator is represented by a three-dimensional vector of functions depending on controls and trajectory:

$$\begin{aligned} S^x_{\lambda,q} \star v(t) &= \frac{\partial \zeta_x}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_x}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_x}{\partial q_3} \star v_3(t) \\ S^y_{\lambda,q} \star v(t) &= \frac{\partial \zeta_y}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_y}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_y}{\partial q_3} \star v_3(t) \\ S^z_{\lambda,q} \star v(t) &= \frac{\partial \zeta_z}{\partial q_1} \star v_1(t) + \frac{\partial \zeta_z}{\partial q_2} \star v_2(t) + \frac{\partial \zeta_z}{\partial q_3} \star v_3(t) \end{aligned} \tag{5.21}$$

5.2.1 Computer Calculations

Let's illustrate some numerical calculations with MAPLE (see Appendix E.3 for code) to evaluate the Stiffness of an Anthropomorphic Manipulator (Fig.5.5) in motion. The equations of motion (eq.5.18) implemented in MAPLE are given by the following formula:

$$\begin{aligned} &> \mathbf{eq} := \mathbf{Multiply}(\mathbf{B}, \mathbf{DDq}) + \mathbf{Multiply}(\mathbf{C}, \mathbf{Dq}) + \mathbf{Multiply}(\mathbf{Fv}, \\ &> \mathbf{Dq}) + \mathbf{G} - \mathbf{Tau} + \mathbf{Multiply}(\mathbf{Transpose}(\mathbf{J}), \mathbf{w}) : \end{aligned} \tag{5.22}$$

To find contributions of external forces applied to the end-effector, we have to calculate the three-dimensional vector h_e. From eq.5.18 we obtain:

$$> \mathbf{h_e} = \left(\mathbf{J^T(q)}\right)^{\mathbf{-1}} \left(\mathbf{B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + f_s(\dot{q}) + g(q)} - \tau\right) \tag{5.23}$$

$$\begin{aligned} &> \mathbf{IJ} := \mathbf{simplify(MatrixInverse(J))} : \\ &> \mathbf{wr} := \mathbf{simplify(Multiply(IJ, -Tau + Multiply(B, DDq)+} \\ &> \mathbf{+Multiply(C, Dq) + Multiply(Fv, Dq) + G)) :} \end{aligned} \tag{5.24}$$

The vector wr, which represents the vector of external forces h_e, has three components $\mathbf{wrj}$ ($j = 1..3$):

$$> \mathbf{wr1} := \mathbf{wr[1]} : \mathbf{wr2} := \mathbf{wr[2]} : \mathbf{wr3} := \mathbf{wr[3]} : \tag{5.25}$$

To obtain the Stiffness Operator we have to derive the three expressions in sense of Gâteaux (Appendix B.1. At the beginning we evaluate the three *difference quotients* for each component of h_e in respect of the variations of trajectory of the three joints $q + \delta q = (q_1 + h_1 v_1, q_2 + h_2 v_2, q_3 + h_3 v_3)$, where $\delta q = hv$, with $v = (v_1, v_2, v_3)$ a vector of the space X (def.4.2) and (h_1, h_2, h_3) a triplet of real numbers. Evaluating limits of *difference quotients* when h_j ($j = 1..3$) tends to 0, we obtain nine *Gâteaux partial derivatives*, one for each component of h_e in respect of each trajectory variation of the three joints.

$$> \mathbf{wrisj} := \mathbf{subs}(\mathbf{q[j]} = \mathbf{q[j]} + \mathbf{hj} * \mathbf{vj(t)}, \mathbf{wri}) : \quad i = 1..3 \quad j = 1..3$$
$$> \mathbf{rapincij} := \mathbf{simplify}((\mathbf{wrisj} - \mathbf{wri})/\mathbf{hj}) : \quad i = 1..3 \quad j = 1..3$$

$$> \mathbf{DiffGatij} := \mathbf{limit}(\mathbf{rapincij}, \mathbf{hj} = \mathbf{0}) : \quad i = 1..3 \quad j = 1..3 \tag{5.26}$$

Finally, the three components of the Dynamical Stiffness Operator (def. 5.21) are given by the following formulas:

$$\begin{aligned} &> \mathbf{Stiffx} := \mathbf{DiffGat11} + \mathbf{DiffGat12} + \mathbf{DiffGat13} : \\ &> \mathbf{Stiffy} := \mathbf{DiffGat21} + \mathbf{DiffGat22} + \mathbf{DiffGat23} : \\ &> \mathbf{Stiffz} := \mathbf{DiffGat31} + \mathbf{DiffGat32} + \mathbf{DiffGat33} : \end{aligned} \tag{5.27}$$

In this way, given a generic trajectory perturbation $v(t) = (v_1(t), v_2(t), v_3(t))$, it is possible to determine the external load perturbation applied to the extremity of the chain which causes it.

Evaluation of Stiffness

Let's consider the system in Fig.5.5 and set values as follows: lengths of arms (cylindrical) $a_i = 10\,m$ $(i = 1..3)$, rays of arms $r_i = 0,1\,m$ $(i = 1..3)$, masses of arms $m_i = 5\,kg$ $(i = 1..3)$, coefficients of attrite and of Coriolis effect $\nu_i = f_{vi} = f_{si} = h = 1\,N/m$ $(i = 1..3)$, masses of motors $m_{mi} = 5\,kg$ $(i = 1..3)$, reduction rapports $kr_{ri} = 5$ $(i = 1..3)$ and elastic constants of actuators $k_j = 1\ N/m^2$ $(j = 1..6)$. We can assume as external load, applied at the extremity of the chain, the vector $h_e = (10, 0, 50)$. We can verify that, choosing a particular trajectory, the Dynamical Stiffness Operator of the system changes in respect to the controls applied.

Let us consider the following vectors of controls. $\lambda(t)$ given by:

$$\begin{cases} \lambda_1(t) = -0,78 + \frac{1}{4}\sqrt{21344,1 - 89,13 \cdot t + 16 \cdot t^2} \\ \lambda_2(t) = 2 - t \\ \lambda_3(t) = 0,78 + \frac{1}{4}\sqrt{19162,45 - 3,43 \cdot t + 4 \cdot t^2} \\ \lambda_4(t) = 1 - \frac{t}{2} \\ \lambda_5(3) = -1,57 + \frac{1}{4}\sqrt{610,33 - 50,26 \cdot t + 16 \cdot t^2} \\ \lambda_6(t) = -t \end{cases} \tag{5.28}$$

and $\lambda'(t)$ given by

$$\begin{cases} \lambda'_1(t) = -0,78 + \frac{1}{4}\sqrt{21229,87 - 2,09 \cdot t + 0,11 \cdot t^2} \\ \lambda'_2(t) = \frac{t}{12} \\ \lambda'_3(t) = 0,78 + \frac{1}{4}\sqrt{19171,58 - 25,13 \cdot e^{\frac{t}{2}} t + 16 \cdot e^t} \\ \lambda'_4(t) = -e^{\frac{t}{2}} \\ \lambda'_5(3) = -1,57 + \frac{1}{4}\sqrt{610,33 - 1507,9 \cdot t + 14400 \cdot t^2} \\ \lambda'_6(t) = -30 \cdot t \end{cases} \tag{5.29}$$

For simplicity of notations we assume

$$\begin{aligned} \omega_1 &= \sqrt{21344,1 - 89,13 \cdot t + 16 \cdot t^2} \\ \omega_2 &= \sqrt{19162,45 - 3,43 \cdot t + 4 \cdot t^2} \\ \omega_3 &= \sqrt{610,33 - 50,26 \cdot t + 16 \cdot t^2} \\ \omega'_1 &= \sqrt{21229,87 - 2,09 \cdot t + 0,11 \cdot t^2} \\ \omega'_2 &= \sqrt{19171,58 - 25,13 \cdot e^{\frac{t}{2}} t + 16 \cdot e^t} \\ \omega'_3 &= \sqrt{610,33 - 1507,9 \cdot t + 14400 \cdot t^2}. \end{aligned}$$

Obviously we have to take in consideration the existence conditions. Conditions for existence of square roots for controls $\lambda_j(t)$ are illustrated in Fig.5.7 and for controls $\lambda'_j(t)$ are illustrated in Fig.5.8. We observe that arguments of square roots are always positive.

We assume that $q(t) = (q_1(t), q_2(t), q_3(t)) = (-\pi/4, \pi/4, -\pi/2)$ ($q_i(t)$ =trajectory of the i-joint, see Fig.5.6) is the vector representing the three-dimensional trajectory of the system.

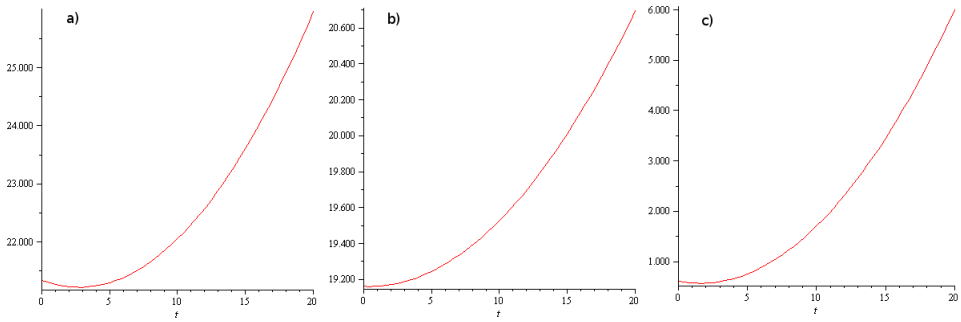

Figure 5.7: *a) Argument of square root of control* $\lambda_1(t)$*. b) Argument of square root of control* $\lambda_3(t)$*. c) Argument of square root of control* $\lambda_5(t)$*.*

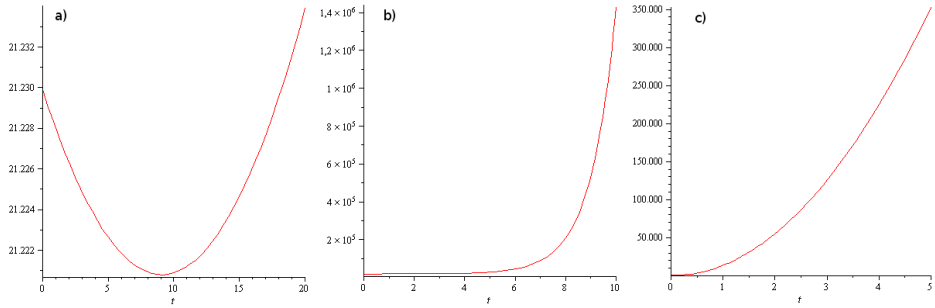

Figure 5.8: *a) Argument of square root of control* ${\lambda_1}'(t)$*. b) Argument of square root of control* ${\lambda_3}'(t)$*. c) Argument of square root of control* ${\lambda_5}'(t)$*.*

The three Dynamical Stiffness Operators $Stiff_\lambda$ and $Stiff_{\lambda'}$ associated associated to controls λ and λ' respectively (def. 5.27) are very complex expressions. As in the bidimensional case, images of a generic trajectory perturbation $v(t)$ by the Stiffness Operators are different functions, and represent the load variations necessary to obtain the perturbation $v(t)$ with controls $\lambda(t)$ and $\lambda'(t)$.

Let us consider the trajectory perturbation

$$v(t) = (v_1(t), v_2(t), v_3(t)) = (\sin(t), \sin(2t), \cos(t))$$

The images by Stiffness Operators relative to controls $\lambda(t)$ and $\lambda'(t)$ are given by:

$$Stiff_\lambda \star v = \begin{bmatrix} t^2 \cdot 10^{-12} 5 \sin(t) + t \cdot \frac{10^{-1}}{4} \left(4 \sin(t) + 10^{-10} \cos(t) + \right. \\ \left. 4 \sin(2t)\right) + \frac{10^{-12}}{4} \left(35,4 \sin(t) - \cos(t) + 10^{10} \sin(2t)\right) \omega_2 + \\ + \frac{10^{-11}}{4} \left(10^{10} \sin(t) + \cos(t)\right) \omega_1 + \\ +6,15 \sin(t) + 56,9 \cos(t) + 36,76 \sin(2t) \\ \\ t \cdot 10^{-1} \left(+1,4 \cos(t) - \sin(t) + 0,5 \sin(2t)\right) + \\ + \frac{10^{-12}}{4} \left(\cos(t) - \sin(t) + \sin(2t)\right) \omega_2 + \\ + \frac{10^{-11}}{4} \left(\cos(t) - 10^{10} \sin(t) + \sin(2t)\right) \omega_1 + \\ + \frac{10^{-1}}{4} \left(1,4 \cos(t) + 2,2 \cdot 10^{-11} \sin(2t)\right) \omega_3 + \\ -19,19 \sin(t) + 23,97 \cos(t) + 110 \sin(2t) \\ \\ t \cdot 10^{-1} \left(2 \sin(t) - 10^{-10} 5,2 \cos(t) - \sin(2t)\right) + \\ + \frac{10^{-12}}{4} \left(\sin(t) - \cos(t) - \sin(2t)\right) \omega_2 + \\ + \frac{10^{-11}}{4} \left(10^{11} 2 \sin(t) - \cos(t)\right) \omega_1 + \\ - \frac{10^{-12}}{4} \left(21,7 \cos(t) + 43,3 \sin(2t)\right) \omega_3 + \\ +90,4 \sin(t) + 1,8 \cos(t) + 228,7 \sin(2t) \end{bmatrix}$$

$$
Stiff_{\lambda'} \star v = \begin{bmatrix}
e^{t/2} \cdot 10^{-12} \left(23,3\sin(t) - 8,1\cos(t) + 10^{11}\sin(2t)\right) + \\
-t^2 \cdot 10^{-15} \left(4,44\sin(t) + 2,78\cos(t)\right) + \\
-t \cdot 10^{-12} \left(10^9 8,3\sin(t) - 2\cos(t) + 0,39\sin(2t)\right) + \\
+\frac{10^{-13}}{4} \left(10^{12}\sin(t) + \cos(t)\right) \omega_1' + \\
+\frac{10^{-13}}{4} \left(14,16\sin(t) - \cos(t) + 10^{12}\sin(2t)\right) \omega_2' + \\
+6,35\sin(t) + 56,9\cos(t) + 36,86\sin(2t) \\
\\
e^{t/2} \cdot 10^{-12} \left(10^{11}\sin(t) - 6,25\cos(t) - 6,25\sin(2t)\right) + \\
+t^2 \cdot 10^{-9} \left(10^{-6}5,6\sin(t) + 2,2\cos(t) + 2,2\sin(2t)\right) + \\
+t \cdot 10^{-12} \left(10^9 8,3\sin(t) - 10^{12}4,2\cos(t) - 9,2\sin(2t)\right) + \\
+\frac{10^{-13}}{4} \left(10^{12}\sin(t) + \cos(t) + \sin(2t)\right) \omega_1' + \\
+\frac{10^{-13}}{4} \left(\cos(t) - \sin(t) + 10^{12}\sin(2t)\right) \omega_2' + \\
+\frac{10^{-12}}{4} \left(10^{12}0,14\cos(t) + 21,7\sin(2t)\right) \omega_3' + \\
-19,39\sin(t) + 23,97\cos(t) + 110\sin(2t) \\
\\
e^{t/2} \cdot 10^{-11} \left(1,3\sin(t) - 62,5\cos(t) + 10^{12}2\sin(2t)\right) + \\
+t^2 \cdot 10^{-9} \left(10^{-4}1,1\sin(t) - 2,2\cos(t) + 0,2\sin(2t)\right) + \\
+t \cdot 10^{-9} \left(10^{-3}9,2\cos(t) - 10^7 1,6\sin(t) + 3,4\sin(2t)\right) + \\
+\frac{10^{-13}}{4} \left(10^{12}2\sin(t) + \cos(t)\right) \omega_1' + \\
+\frac{10^{-13}}{4} \left(2\sin(t) - \cos(t) - 10^{12}2\sin(2t)\right) \omega_2' + \\
-\frac{10^{-11}}{4} \left(2,17\cos(t) + 4,33\sin(2t)\right) \omega_3' + \\
+90,8\sin(t) + 1,8\cos(t) + 228,9\sin(2t)
\end{bmatrix}
$$

Values of dynamical stiffness in presence of the trajectory perturbation $v(t) = (\sin(t), \sin(2t), \cos(t))$ associated to $\lambda(t)$ and $\lambda'(t)$ are represented in Fig.5.9 and 5.10 respectively. Blue lines represent components x of stiffness, red lines represent components y and green lines components z.

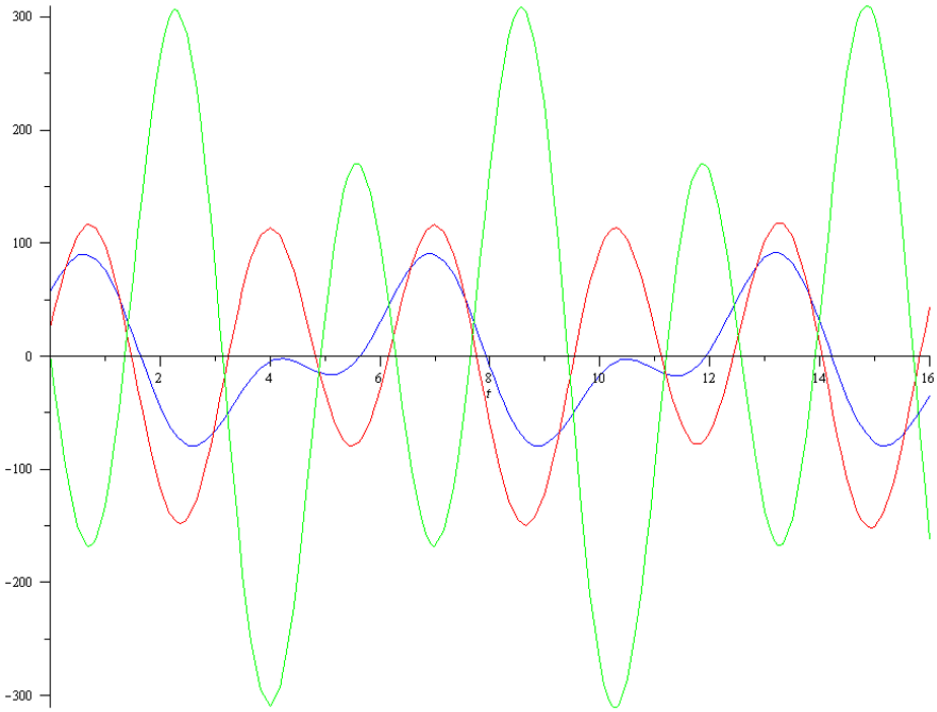

Figure 5.9: *Dynamical Stiffness of the anthropomorphic manipulator* ($Stiff_x \star v(t)$ =*blue line,* $Stiff_y \star v(t)$ =*red line,* $Stiff_z \star v(t)$ =*green line) with trajectory perturbation* $v(t) = (\sin(t), \sin(2t), \cos(t))$ *and controls* $\lambda(t)$.

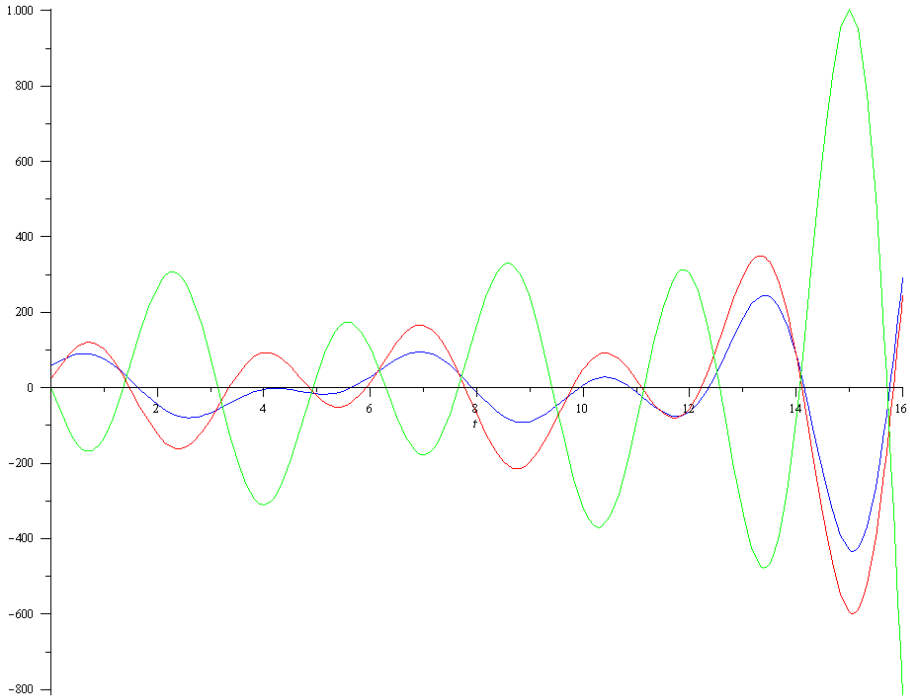

Figure 5.10: *Dynamical Stiffness of the anthropomorphic manipulator* ($Stiff_x \star v(t)$ =*blue line,* $Stiff_y \star v(t)$ =*red line,* $Stiff_z \star v(t)$ =*green line) with trajectory perturbation* $v(t) = (\sin(t), \sin(2t), \cos(t))$ *and controls* $\lambda'(t)$.

Chapter 6

The Dynamic Compliance Operator

6.1 Mathematical Background

In the previous chapter we have generalized the concept of stiffness to the case of a system in motion. The concept of Dynamic Stiffness can be useful in the treatment of many mechanical problems, as, for example, the estimation of the force generated by the system when it commits an error in terms of its trajectory, fundamental in breakable object manipulation. On the other and, the knowledge of a measure of an extended concept of compliance would be very important in estimating the deviation from a given trajectory, when the perturbation amount is evaluable, foreseeable or estimable.

Unfortunately, the calculation of an explicit expression for the compliance functional is not so easy as the stiffness is, since the representation of the phenomenon is in terms of impedance (and not in admittance ones). This implies that even if it is possible to define a "length map"

corresponding to the load map, and a compliance functional as the map

$$C_{\lambda,x} : P \ni f(t) \mapsto v(t) \in X \tag{6.1}$$

which associates an external force variation $f(t)$ imposed to the system with the solution of the following Cauchy problem:

$$\begin{cases} \ddot{v} + \xi \dot{v} - g(t)v = f(t) \\ \dot{v}(0) = 0 \\ v(0) = 0 \end{cases} \tag{6.2}$$

where the boundary conditions are set considering that the trajectory followed by the system before an eventual perturbation is "unperturbed". Problem 6.3 is not generally solvable with an explicit formula, because a second order time varying coefficient differential equation, as it is easy to prove, is equivalent to a Riccati's equation [10]. In order to completely determine the compliance operator, the knowledge of all the images of the L^2 basis of P, for all $p \in P$, should be obtained. This implies to solve a scheme of Riccati's equations, one for each element of the L^2 basis and for each fiber of the bundle P.

6.1.1 Restriction to a Finite Dimensional Space

The *Finite Elements Theory* permits to find a very good approximation of our problem in finite dimensional spaces.

In particular *Galerkin Approximation Method* [23] permits to project our differential problem in a finite-dimensional domain and to find there a solution arbitrarily close in norm to the solution of the infinite-dimensional problem.

Restricting to an appropriate finite dimensional domain, the inversion of the operator S becomes possible.

We can find (with complex mathematical computations) the minimum dimension m of the restricted domain, necessary to perform an error lesser than a chosen tolerance ϵ.

To apply the Galerkin method it is necessary to convert our differential problem in a variational problem [2], [23].

The problem we consider has Neumann conditions:

$$\begin{cases} \ddot{v} + \xi\dot{v} - g(t)v = f(t) \\ \dot{v}(0) = 0 \\ v(0) = 0 \end{cases} \tag{6.3}$$

where $\xi = \nu/\mu$, and $g(t)$ is a time-function depending on controls and trajectory. For example, in the eq. 4.12 $g(t)$ has the following expression:

$$g(t) = 2k_1(x(t) - \lambda_1) + 2k_2(l - x(t) - \lambda_2). \tag{6.4}$$

The problem 6.3 can assume the variational form:

$$\int_I u(t)\ddot{v}dt + \int_I \xi u(t)\dot{v}dt - \int_I g(t)u(t)vdt = \int_I f(t)u(t)dt \tag{6.5}$$

The solution $v(t)$ of the problem 6.5 is defined on a limited interval I (for example $I = [0, 2\pi]$). It belongs to the Hilbert space:

$$H = \left\{v \in H^2(I) | v(0) = v(2\pi) = 0, \dot{v}(0) = 0\right\} \tag{6.6}$$

and it represents the *weak solution* for the problem 6.3.

It is a note fact that problem 6.3 has a unique solution, consequently the weak one represents exactly the solution we were searching for. Since

$$a(u, v) = -\int_I \dot{v}\dot{u}dt + \int_I \xi u\dot{v}dt - \int_I g(t)uvdt = \int_I f(t)udt \tag{6.7}$$

is a bilinear form continuous and coercitive we can apply Lax-Milgram Theorem [23], which ensures existence and uniqueness of the weak solution.

Since we are approximating with Galerkin method, we can apply the following useful theorem [23]:

Theorem of Cea: If conditions of Lax-Milgram are satisfied and c_1 and c_2 are the constants of limitation and coercitivity respectively of the bilinear form, then the following inequality holds:

$$\|v - v_m\|_H \leq \frac{c_1}{c_2} \min_{u \in H_m} \|v - u\|_H . \tag{6.8}$$

Where H_m is the finite-dimensional space of dimension m considered in the Galerkin approximation and v_m is the solution of variational problem in H_m.

If the form is symmetric the estimation 6.8 can be better (see appendix C for proof):

$$\|v - v_m\|_H \leq \sqrt{\frac{c_1}{c_2}} \min_{u \in H_m} \|v - u\|_H . \tag{6.9}$$

Luckily our differential form can be made symmetric by multiplying it by $z(t) = e^{\xi t}$. From 6.3 we have:

$$\frac{d(z(t)\dot{v})}{dt} - g(t)z(t)v = f(t)z(t)$$

inverting sign and simplifying notations the problem is reduced in Sturm-Liouville form:

$$-\frac{d(p(t)\dot{v})}{dt} + q(t)v = -h(t) \tag{6.10}$$

where:

- $p(t) = z(t) > \beta$.
- $h(t) = z(t)f(t)$.

- $q(t) = g(t)z(t) = z(t)[2k_1(x - \lambda_1) + 2k_2(l - x - \lambda_2)]$ where the quantity in brackets is always positive or zero.

The associated bilinear form is symmetric:

$$a(u, v) = \int_I p(t)\dot{v}\dot{u}dt + \int_I q(t)uvdt \tag{6.11}$$

The associated weak solution is the function $v(t)$ solving the following equation:

$$a(u, v) = -\int_I h(t)vdt \tag{6.12}$$

The constants c_1 and c_2 of continuity and coercitivity can be easily evaluated:

$$|a(u, v)| \leq \max|p(t)| \int_I \dot{v}\dot{u}dt + \max|q(t)| \int_I uvdt \leq$$

$$\leq \max|p(t)| \|u\|_H \|v\|_H + \max|q(t)| \|u\|_H \|v\|_H .$$

$$a(v, v) = \int_I p(t)\dot{v}^2dt + \int_I q(t)v^2dt \geq \beta \|v\|_H^2 .$$

We have that $c_1 = \max(|p(t)|+|q(t)|)$ and $c_2 = \min(z(t))$. Weierstrass theorem ensures that the functions $p(t)$, $q(t)$ and $z(t)$ have maximum and minimum on $I = [0, 2\pi]$.

To apply the formula in diseq. 6.9 it is necessary to evaluate $\min_{u \in H_m} \|v - u\|_H$. The minimum of the distance of the solution v from a generic vector u of the space H_m, generated by the first $2m + 1$ vectors of basis given in eq. 4.16, is simply the difference between v and its projection on H_m.

This distance can be easily evaluated by the following Lemma.

Lemma: If $\sup_{t\in[0,2\pi]}|\dot{v}(t)|=K$, $\sup_{t\in[0,2\pi]}|\ddot{v}(t)|=T$ and $\sup_{t\in[0,2\pi]}|\dddot{v}(t)|=Q$ then:

$$\int_0^{2\pi} v(t)sin(nt)dt \leq \frac{2\pi K}{n} \ ; \ \int_0^{2\pi} v(t)cos(nt)dt \leq \frac{2\pi K}{n};$$

$$\int_0^{2\pi} \dot{v}(t)sin(nt)dt \leq \frac{2\pi T}{n} \ ; \ \int_0^{2\pi} \dot{v}(t)cos(nt)dt \leq \frac{2\pi T}{n};$$

$$\int_0^{2\pi} \ddot{v}(t)sin(nt)dt \leq \frac{2\pi Q}{n} \ ; \ \int_0^{2\pi} \ddot{v}(t)cos(nt)dt \leq \frac{2\pi Q}{n}.$$

Dim: Proof for all disequations is analogue, so we illustrate only the first one. Subdividing the interval $[0;2\pi]$ in n equal subintervals $I_1,..,I_n$, by Lagrange we have $|v(x)-v(x_i)| \leq K(1/n)$ on $I_i=[x_i,x_{i+1})$, from which $\int_{I_i} v(t)sin(nt)dt \leq \frac{2\pi K}{n^2}$.

We obtain the distance between v and its projection $\hat{v}$ on H_m:

$$\|v-\hat{v}\|_H =$$

$$=\sqrt{\sum_m^{\infty}\left[(c_n^{H^2})^2+(s_n^{H^2})^2\right]+\left[(\dot{c}_n^{H^2})^2+(\dot{s}_n^{H^2})^2\right]+\left[(\ddot{c}_n^{H^2})^2+(\ddot{s}_n^{H^2})^2\right]}$$

$$\leq \frac{2\pi(K+T+Q)}{\sqrt{\pi}}\sqrt{3\sum_m^{\infty}\frac{1}{n^4}} \tag{6.13}$$

Recalling that when m is big $\sum_m^{\infty}\frac{1}{n^4} \leq \sum_m^{\infty}\frac{1}{n^2(n^2-1)} = \sum_m^{\infty}\frac{1}{n^2} - \frac{1}{n^2-1} = \frac{1}{m^2}$, we have:

$$\|v-\hat{v}\|_H \leq \frac{\sqrt{12\pi}(K+T+Q)}{m}.$$

Given controls we can determine the sup of the derivatives of v in the interval (see Appendix D for proof).

Substituting in 6.8 we have the following evaluation:

$$\|v - v_m\|_H \leq \sqrt{\frac{12\pi c_1}{c_2}} \frac{(K + T + Q)}{m}. \tag{6.14}$$

If maximum tolerable error is ϵ, we can restrict the analysis of our differential problem to a finite dimensional space with dimension at least m (the least integer respecting the following inequality):

$$m > \sqrt{\frac{12\pi c_1}{c_2}} \frac{(K + T + Q)}{\epsilon}. \tag{6.15}$$

As stressed before, to consider a finite-dimensional domain is very useful, because we can invert the functional S and deal with compliance instead of stiffness.

6.1.2 Computer Calculations

Let us consider the system of Fig. 3.2, representing a couple of agonist-antagonist springs with monodimensional trajectory.

System in motion in presence of perturbations (eq. 4.12):

$$\mu\ddot{x} + \nu\dot{x} + k_1(x - \lambda_1)^2 - k_2(l - x - \lambda_2)^2 = L_{x,\lambda}(t)$$

Using the software MAPLE we have implemented a program which evaluates, once decided the values of parameters and the load perturbation, the minimum dimension m of the space necessary to calculate the corresponding variation of trajectory for the system 4.12, committing an error lesser than a chosen ϵ (see Appendix E.5 for code).

We have set $l = 10\,m$, $F_1 = 4\,N$, $\mu = 1\,kg$, $\nu = 0,1\,N/m$, $k_1 = k_2 = 0,01\,N/m^2$, let us suppose that system 4.12 is in equilibrium in $x = 5\,m$ for $t = 0$. For $t < 2$ we have set controls $\lambda_1 = 3$ and $\lambda_2 = 4$; at $t = 2$ controls assume values $\lambda_1 = 4$ and $\lambda_2 = 5$ definitively.

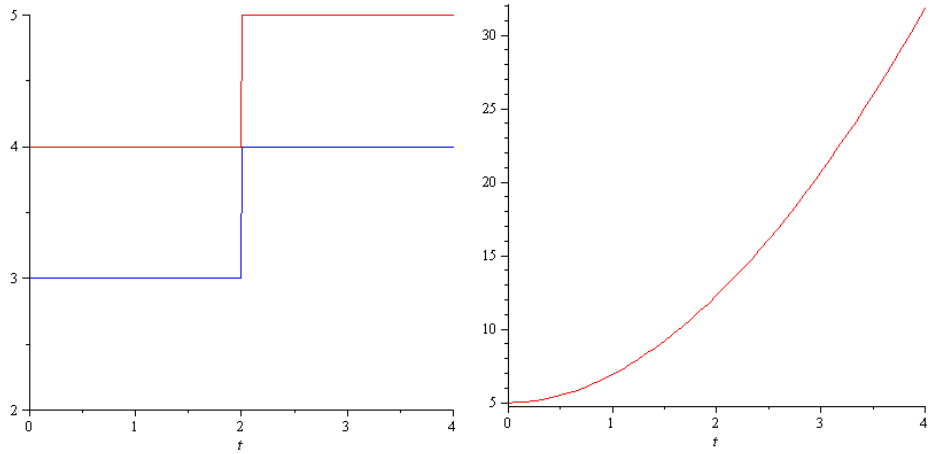

Figure 6.1: *a) Plot of controls applied to the system in motion. The blue line is $\lambda_1(t)$, the red line is $\lambda_2(t)$. b) Trajectory of the system.*

We suppose that the system is interested by a load perturbation $f(t) = \sin(t)$ in the time-interval $I = [0, \frac{\pi}{4}]$.

The trajectory perturbation, consequence of the load perturbation f, is represented by the solution of the Cauchy Problem in 6.3. In general this solution can't be find, but in this case we have chosen piecewise constant controls, and the differential equation can be easily solved. Solution is represented by the following continuous function (Fig.6.1):

$$v(t) = \begin{cases} -0,3e^{-1,17t} + 0,7e^{0,17t} + \\ -0,4\cos(t) - 0,5\sin(t) & t < 2 \\ \\ 0,8e^{(0,8-1,4t)} + 6,5e^{(-2,8+0,4t)} + \\ +0,1e^{(-0,8+0,4t)} - 0,7e^{(2,8-1,4t)} & t \geq 2 \\ -0,3\cos(t) - 0,5\sin(t) \end{cases} \tag{6.16}$$

To validate results presented in chapter 6, we can find an approximation of the solution of the problem by the compliance operator restricted

to a finite domain (which dimension can be evaluated by formula 6.15).

We will verify that it is closer (in norm) to the solution $v(t)$ (eq.6.16) than a chosen tolerance ϵ.

We have chosen $\epsilon = 1$ as maximum tolerable error.

Inserting values in the program implemented in MAPLE, we can evaluate the constants of *continuity* and *coercitivity*, c_1 and c_2:

$$\begin{array}{cc} c_1 = \max(e^{\frac{\nu t}{\mu}} + |g(t)|e^{\frac{\nu t}{\mu}}) & c_2 = \min(e^{\frac{\nu t}{\mu}}) \\ c_1 = 0,0649 & c_2 = 1 \end{array} \tag{6.17}$$

The sup of $|\dot{v(t)}|$, $|\ddot{v(t)}|$ and $|\ \dddot{v}\ (t)|$ (K, T and Q respectively) can be evaluated applying *Gronwall's Lemma* (Appendix D):

$$K = 0,62972 \qquad T = 4,07041 \qquad Q = 1,44482$$

Applying the formula 6.15 it is possible to find the least dimension of the restricted domain, necessary to evaluate the compliance with an error lesser than ϵ. This dimension is given by the least integer greater than:

$$\sqrt{\frac{12\pi c_1}{c_2}\frac{(K+T+Q)}{\epsilon}} = 9,61202$$

So the minimum dimension of the space necessary to have a tolerance lesser than $\epsilon = 1$ is $n = 10$ and we have to keep in consideration at least the first 10 vectors of the basis 4.16, in order to obtain an error lesser than 1.

In order to calculate the matrix associated to the Compliance Operator in the space restricted, it is necessary to choose an orthonormal basis for the space H_{10} (space H defined in 6.6, of dimension 10). The vectors of this basis must satisfy conditions in 6.6 (null on boundary and $\dot{v}(0) = 0$). To buit a basis with these characteristics I have taken in considerations a basis of belts, fig.6.2, and to orthonormalize the vectors I have applieded

the Grahm-Schmidt Method, fig.6.3 (see Appendix E.4 for MAPLE code used to orthonormalize the basis).

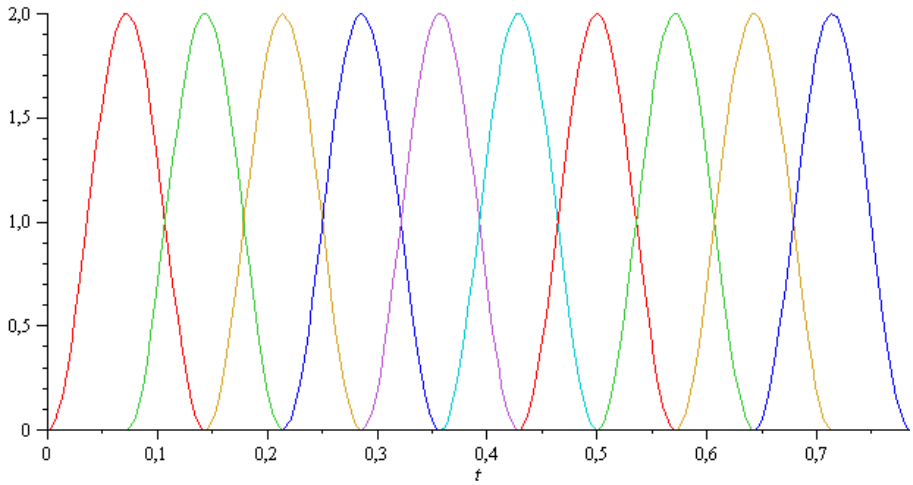

Figure 6.2: *Basis of belts in* $I = [0, \frac{\pi}{4}]$.

Calculating the Stiffness related to the 10-dimensional space, we obtain the square matrix S_{10} (10×10). Its determinant is $7,39439 \times 10^{32}$ (different from zero), so it can be inverted.

Inverting matrix S_{10} we obtain the matrix C_{10}, representing the compliance operator of the system in the finite-dimensional space.

Determining the vector of the first 10 coefficients of the perturbation $f(t) = \sin(t)$ (in respect to basis 6.3), evaluating its image by C_{10} and calculating its representation in basis 6.3, we obtain a function $\hat{v}(t)$ "close" to $v(t)$.

$$f(t) = \sin(t) \longrightarrow \hat{f}(t)$$

$$C_{10} \circ \hat{f}(t) \longrightarrow \hat{v}(t)$$

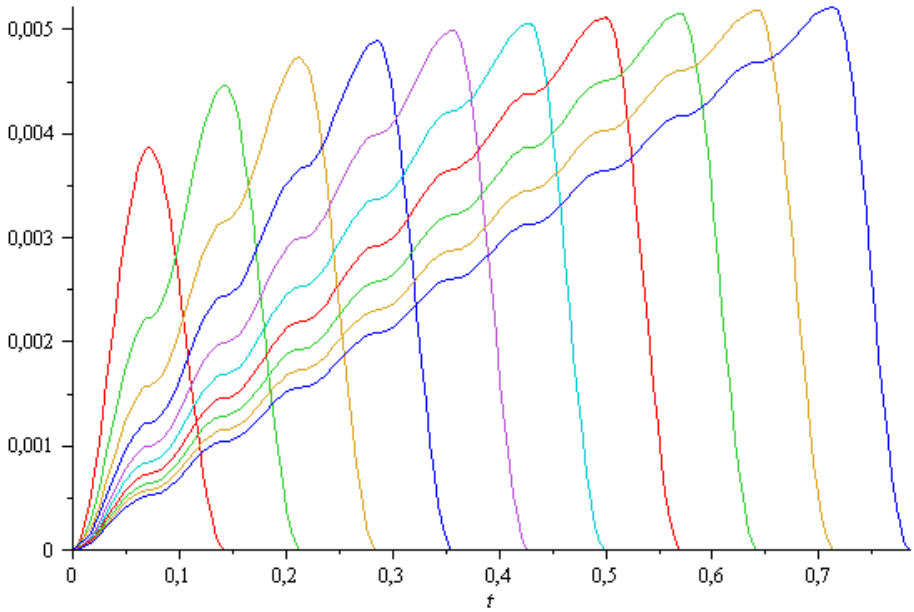

Figure 6.3: *Basis of belts ortonormalized with Grahm-Schmidt Method in* $I = [0, \frac{\pi}{4}]$.

$$\|\hat{v}(t) - v(t)\|_H = \left(\int_0^{\frac{\pi}{4}} (\hat{v} - v)^2 + (\dot{\hat{v}} - \dot{v})^2 + (\ddot{\hat{v}} - \ddot{v})^2 dt \right)^{\frac{1}{2}}.$$

Calculating this distance we obtain:

$$\|\hat{v}(t) - v(t)\|_H = 0,383822$$

which is a value lesser than $\epsilon = 1$.

Chapter 7

Applications

7.1 Electroactive Materials and Fibers

One of the aims of this thesis is to propose a feasible technical solution able to implement the Feldman's mechanical characteristic in a biomimetic way. In addition to the requirements on the non-linearity of the mechanical (force versus strain) characteristic, we would satisfy properties such as space-saving, lightness and programmability of the device to mime several different muscle actions. In order to do it, Electro-active polymers have been considered. Electro-active materials such as electro-strictive polymers can convert directly electrical energy into mechanical energy. They can work as actuators in the form of capacitors with the dielectric made of electro-active polymers. When an electrical stimulus is applied, the electrical field generates a force that deforms the material [7]. Cylindrical fibres have been built by these technologies, with one electrode placed along the axis and the other one distributed over the external surface. These fibres show good elastic properties and modify their rest length when they receive electrical stimuli. Unfortunately, since they change shape without changing volume, the desired Feldman's quadratic

law (eq. 2.2) cannot be obtained directly but only approximately by combining the effects of many fibres.

7.2 Control of Biomimetic Chains

7.2.1 Control Strategies

A complete control of a kinematic chain (i.e. including compliance and stiffness) can be reached taking in consideration the set of all actuators. By completing the set of state variables as exposed at the beginning of chapter 3, i.e. $z = (q|c)$, we have obtained a full rank local function $Z = h(L)$ from control space L to state space Z. This function stands at the basis of the feed-forward control of the kinematic chain linked to the actuators. Chosen a class of mechanical characteristics for a certain muscle to be mimed, and given the electroactive bundle compatible with the mechanical requirements, a Peripheral Control Unit (PCU) which realizes the recruitment has to be designed. In order to reply the Feldman's theory, this unit has λ (provided by a Central Control Unit, CCU) and x (provided by a suitable sensing system) as inputs. Given these two variables, the PCU has to compute the exact configuration of the bundle (the set of fiber which have to be activated, i.e. the ones whose rest lengths have to be smaller than x) according to performances required for the controlled muscle. The central unit, which computes all the central variables λs for each muscle involved in the kinematic chain to set position and compliance, keeps enough computational resources since the computation of the "details" on the fiber status is demanded to peripheral control units. This is the basis of the control for a biomimetic robot based on the Feldman's equilibrium point theory. This architecture can be completed by a Central Memory Unit which can store the λs evolution during certain fundamental movements of the mechanical chain. This

unit partially replicates the role of the biological cerebellum in managing habitual movements, i.e. the movements that a subject performs without keeping particular attention on what he is doing. The most accredited theories [38], [22] demand the choice of these habitual trajectories to the minimization of variational functionals, such as the minimum jerk and the minimum torque change hypotheses respectively

$$J = \sum_{t_0}^{t} \int_{k=1}^{n} \left(\frac{d^3 q_k}{dt^3} \right)^2 dt \quad T = \sum_{t_0}^{t} \int_{k=1}^{n} \left(\frac{d^3 \tau_k}{dt^3} \right)^2 dt \tag{7.1}$$

where q_k are the geometrical state variables and τ_k the torques at the joints of the controlled chain. An interesting field of investigation is the characterization of movements (including compliances) in terms of λs, with particular care to the ones satisfying properties 7.1.

7.2.2 PM Actuators and Progressive Recruitment

The PM (Pseudo Muscular) actuators realized with dielectric elastomers have as common property that when deformed under the action of the electric field, their volume unchanges. Holding account of such property it is possible to gain the characteristic force-lengthening of such actuators in a range of deformation in which the Young's modulus E of the material can be considered constant. The relation stress-strain for a linearly elastic body, isotropic and homogeneous, subject to axial stress (according z) is given by:

$$df_b = EA(z)\frac{dz}{z}. \tag{7.2}$$

If a cylindrical body is considered, the area of a section, related to the length z, is given by:

$$A(z) = \frac{V_0}{z} \tag{7.3}$$

where V_0 represents the volume of the fiber. Thus, due to the iso-volumetric hypothesis, the infinitesimal force exerted by a portion of the

fiber is:

$$df_b = EV_0 \frac{dz}{z^2}. \tag{7.4}$$

In order to obtain f_b, by integrating on the interval $[\mu, x]$ we have:

$$f_b = \int_\mu^x EV_0 \frac{dz}{z^2} = EV_0 \left(\frac{1}{\mu} - \frac{1}{x} \right) u(x - \mu) \tag{7.5}$$

where $u(\cdot)$ represents the Heaviside function. In Fig. 7.1 the length-force curves for an acrylic elastomer fiber driven by different electric fields are plotted. Dotted lines represent the experimental data, while solid lines represent the theoretical data derived by Eq. 7.5.

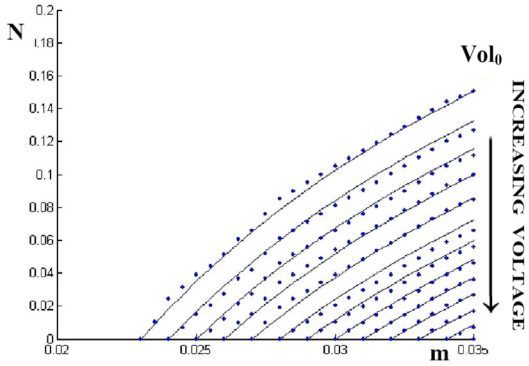

Figure 7.1: *Length-force curves for an acrylic elastomeric fiber, with $Y = 0.07$ MPa, $V_0 = 1.3800\,10^{-7}$ m^3, $\mu = 0.023$ m (unstretched)* driven by different electric fields (E). Force is measured in Newton. Dotted lines represent the experimental data, while solid lines represent the theoretical data.

When many collinear fibers, a set I, having actual length x, are grouped in a bundle B, the resultant supplied force is

$$F_B = \sum_{i \in I^*} f_{b_i} = \sum_{i \in I^*} EV_0 \left(\frac{1}{\mu_i} - \frac{1}{x} \right) u(x - \mu_i) \tag{7.6}$$

where I^* is the set of active fibers, the ones for which $x > \mu_i$ holds. It is worth noting that the global behaviour of the bundle can be modified

by selecting a suitable activation order for the fibers I^*. Hence it is also possible to approximate Feldman's muscle model in Eq. 2.2 by using a fibers bundle. In other terms it is possible, for a chosen tolerance ϵ, to define an activation order that satisfies the following relation

$$\|F - F_B\|_{C^1} = \sup_{x\in\Omega} |F - F_B| + \sup_{x\in\Omega} \left| \frac{\partial F}{\partial x} - \frac{\partial F_B}{\partial x} \right| =$$

$$= \sup_{x\in\Omega} \left| k(x-\lambda)^2 - \sum_{i\in I^*} EV_0 \left(\frac{1}{\mu_i} - \frac{1}{x} \right) u(x-\mu_i) \right| +$$

$$+ \sup_{x\in\Omega} \left| 2k(x-\lambda) - \sum_{i\in I^*} EV_0 \left(\frac{1}{x^2} \right) u(x-\mu_i) \right| < \epsilon \tag{7.7}$$

($x > \mu_i$, $i \in I^*$), where Ω is an union of open intervals where F_B is differentiable. The choice of the $C_1(\Omega)$ norm, which is not defined in the points x where a single fiber of the bundle is activated, ensures the possibility of obtaining a position and stiffness control for a kinematic chain according to the required non linearity of the muscle mechanical characteristic.

Results shown in Fig. 7.2 have been obtained by combining the effects of 21-fibers to join a single pseudomuscular actuator. By exciting all the fibers simultaneously, the characteristic shifts when the global rest length changes (by decreasing μ_i with $i \in I^*$), as for biological muscles.

7.2.3 Electroactive Polymers Actuators

Among the perspectives of application, a beautiful implementation could be the realization of a kinematic chain actuated by electroactive polymers and controllable in position and in stiffness both in the static and in the dynamic case. A prototype coul be represented by an artificial muscle composed by Rolled Fibres [7], i.e. elastometer actuators of cylindrical shape (Fig.7.3).

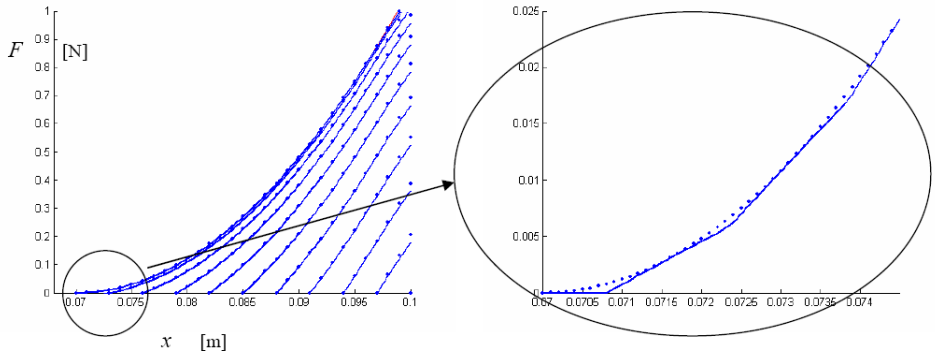

Figure 7.2: *Feldman's muscle model approximation obtained using* 21 *fibers with effective Young's modulus Y=0.07 MPa. Force is measured in Newton. The solid lines represent the sum of* 7.5*, while dotted lines represent experimental data.*

The alimentation and the control of these Rolled could be realized by ultra-micro integrated amplifiers (Fig. 7.4). A necessary characteristic of the polymeric bundle would be the possibility to differentiate the recruitment of the fibres.

In figure 7.5 is illustrated a prototype of robotic arm realized with electro active elastomeric actuators by EMPA (Swiss Federal Laboratories for Materials Testing and Research, Dubendorf, Switzerland).

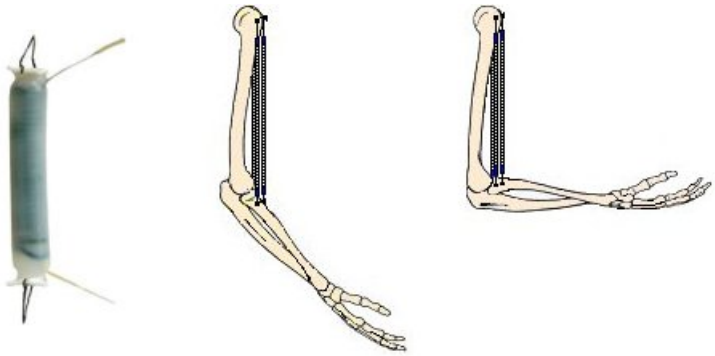

Figure 7.3: *Schema of an Artificial musculo-skelethal system, realized with a bone structure of resina and electro active polimeric actuators.*

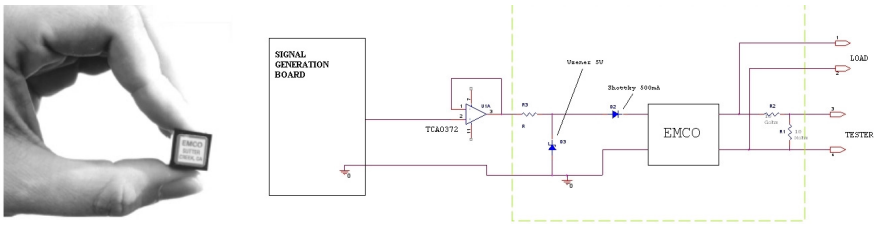

Figure 7.4: *Ultra-Miniatur DC to HVDC converter and its circuit.*

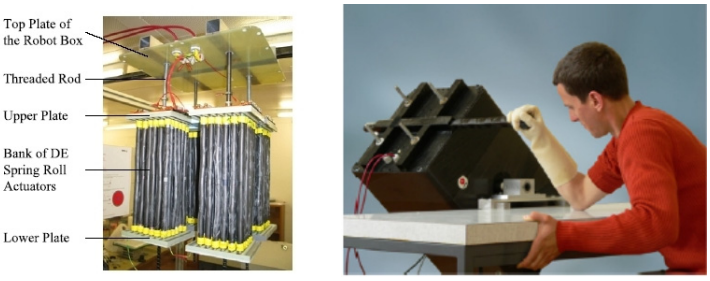

Figure 7.5: *Prototype of robotic arm composed by 4 bundles of actuators, each of them composed by 64 fibers.*

Conclusions

The main contribution of this work would be the attempt to give a physical-mathematical interpretation of particular characteristics of an amazing biological phenomenon, like the movement is, by the definition of two new operators (Dynamic Stiffness and Compliance) devoted to the analysis of properties, characteristics and behaviours of biological muscles.

The effort of abstraction and synthesis carried out in this thesis has revealed new interesting relations between the characteristic of a muscle and its behaviour in presence of external perturbations.

The way to understand the complex mechanisms related to motion control is still a long steep path, but I see, around me, many people performing little steps, and the way goes on, faster and faster.

Appendix A

Elements of Functional Analysis

A.1 Banach and Hilbert Spaces

Definition: *Let X be a vectorial space on $\mathbb{R}$ or $\mathbb{C}$. A norm on X is a function $\|\cdot\| : X \longrightarrow [0, \infty[$ with thw following properties:*

(i) $\|x\| = 0 \Longleftrightarrow x = 0$;

(ii) $\|\lambda x\| = |\lambda| \, \|x\| \ \forall x \in X$ *and* $\forall \lambda \in \mathbb{R}$ $(\lambda \in \mathbb{C})$;

(iii) $\|x + y\| \leq \|x\| + \|y\| \ \forall x, y \in X$.

The Space $(X, \cdot)$ is called Normed Space and is a metric space with the induced distance $d(x, y) = \|x - y\|$; if such metric space is complete, $(X, \cdot)$ is called Banach Space.

Definition: *Let H be a vectorial space on $\mathbb{C}$. An inner product on H is an application $(\cdot, \cdot) : H \times H \longrightarrow \mathbb{C}$ with the following properties:*

(i) (x, x) *is real and non-negative,* $(x, x) = 0 \Longleftrightarrow x = 0$;

(ii) $(x, y) = \bar{(y, x)}$ $\forall x, y \in H$;

(iii) $x \mapsto (x, y)$ *is linear* $\forall y \in H$.

The Space H with the inner product $(\cdot, \cdot)$ is called Pre-Hilbert Space or Space with Inner Product.

Proposition: *Let H be a space with inner product $(\cdot, \cdot)$. Then*

$$\|x\| = \sqrt{(x, x)}$$

is a norm on H, induced by the inner product.

If a Pre-Hilbert Space H is complete in respect to the norm induced by the inner product, it is called *Hilbert Space*.

A.2 SONC and Fourier basis

Definition: *A set $\{e_\alpha\}_{\alpha \in A}$ of elements of an Hilbert Space H is called Orthonormal System (SON), in respect to the inner product $(\cdot, \cdot)_H$ of the space H, if results:*

$$(e_\alpha, e_\beta)_H = \delta_{\alpha\beta} = \begin{cases} 1 & se \quad \alpha = \beta \\ 0 & se \quad \alpha \neq \beta \end{cases} \qquad \forall \alpha, \beta \in A. \tag{A.1}$$

Definition: *Let H be an Hilbert Space and let $\{e_\alpha\}_{\alpha \in A}$ be a SON in H. We say that $\{e_\alpha\}_{\alpha \in A}$ is complete if $[\bar{\{e_\alpha\}_{\alpha \in A}}] = H$.*
In other words if the finite linear combinations of elements of $\{e_\alpha\}_{\alpha \in A}$ are dense in H in respect of the norm on $\|\cdot\|_H$, induced on H by its inner product. In this case we say that $\{e_\alpha\}_{\alpha \in A}$ is a SONC.

The Trigonometric System:

$$\left\{ \frac{1}{\sqrt{2\pi}}, \frac{\cos(nx)}{\sqrt{\pi}}, \frac{\sin(nx)}{\sqrt{\pi}} \right\}_{n\in\mathbb{N}^*} \tag{A.2}$$

is a SONC in the Hilber space $H^l(-\pi,\pi)$ in respect to the norm $\|\cdot\|_2$ (norm of Sobolev Space $W^{l,2}(-\pi,\pi)$, see def. **??**).

A.3 Sobolev Spaces

Definition: *If $\Omega \in \mathbb{R}$ is an open connected set, then $L^q(\Omega)$, $1 \leq q < \infty$, is the set of all measurable functions $u(x)$ in Ω such that the norm*

$$\|u\|_{q,\Omega} = \left(\int_\Omega |u(x)|^q dx \right)^{1/q} \tag{A.3}$$

is finite.

Proposition: *$L^q(\Omega)$ is a Banach Space.*

Definition: *Suppose that $u \in L^p(\Omega)$ and suppose that exist weak derivatives $\partial^\alpha u$ for any α with $|\alpha| \leq l$ (all derivatives up to order l), such that*

$$\partial^\alpha u \in L^p(\Omega), \quad |\alpha| \leq l.$$

Then we say that $u \in W^{l,p}(\Omega)$.

Definition: *The standard norm in $W^{l,p}(\Omega)$ is so defined:*

$$\|u\|_{W^{l,p}(\Omega)} = \left(\int_\Omega \sum_{|\alpha|\leq l} |\partial^\alpha u|^p dx \right)^{1/p}. \tag{A.4}$$

$\forall p < \infty$ $W^{l,p}(\Omega)$ is a Banach space. In particular we observe that $W^{0,p}(\Omega) = L^p(\Omega)$.

Important Observation: *If $p = 2$, the space $W^{l,2}(\Omega)$ is an Hilbert Space with the inner product*:

$$(u, v)_{W^{l,2}(\Omega)} = \int_\Omega \sum_{|\alpha| \leq l} |\partial^\alpha u(x)| \, |\partial^\alpha \bar{v}(x)| dx. \tag{A.5}$$

Since $W^{l,2}(\Omega)$ are Hilbert Spaces, another notation is often used:

$$W^{l,2}(\Omega) = H^l(\Omega).$$

In particular $W^{1,2}(\Omega) = H^1(\Omega)$ with the inner product:

$$(u, v)_{W^{l,2}(\Omega)} = \int_\Omega |u(x)| \, |\bar{v(x)}| + |\partial u(x)| \, |\partial \bar{v(x)}| dx. \tag{A.6}$$

and the induced norm:

$$\|v\|_2 = \sqrt{\int_\Omega (v(x))^2 + (\partial v(x))^2 dx}. \tag{A.7}$$

Appendix B

Fréchet and Gâteaux Derivative

The *Fréchet differential* is a derivative defined on Banach spaces. The *Gâteaux derivative* is a generalization of the concept of directional derivative in differential calculus.

Definition 1: (Fréchet differential) *let L be a functional on an open domain D in a normed space X, and having range in a normed space P. If for fixed $x_0 \in D$ and each $v \in T_{x_0}$ (tangent space of X in x_0) there exists a functional $\delta L(x_0, v) \in \mathcal{L}(T_{x_0}, T_{L(x_0)})$ which is linear and continuous in respect on v such that*

$$\lim_{\|v\| \to 0} \frac{\|L(x_0 + v) - L(x_0) - \delta L(x_0, v)\|}{\|v\|} = 0 \tag{B.1}$$

then $L(x, v)$ is said to be Frechet differentiable at x_0 and $\delta L(x_0, v)$ is said to be the Frechet differential of L at x_0 for an increment v.

Definition 2: (Gâteaux partial derivative) *Let L be a functional on*

an open domain D in a normed space X, and having range in a normed space P. If for fixed $x_0 \in D$ and each $w \in T_{x_0}$ (tangent space of X in x_0) there exists a functional $\delta L(x_0, hw) \in \mathcal{L}(T_{x_0}, T_{L(x_0)})$ which is linear and continuous in respect on w such that

$$\lim_{h \to 0} \frac{\|L(x_0 + hw) - L(x_0) - \delta L(x_0, hw)\|}{|h|} = 0 \tag{B.2}$$

Expression B.2is a real function variable limit, which can be effectively calculated, when it is possible.

If a functional L is *Fréchet differentiable*, setting $wh = v$ we obtain that $h \to 0$ if and only if $\|v\| \to 0$:

$$\lim_{h \to 0} \frac{\|L(x_0 + hw) - L(x_0) - \delta L(x_0, hw)\|}{\|w\|\,|h|} = 0 \tag{B.3}$$

for all $v \in T_{x_0}$. Therefore $L : X \longrightarrow P$ is also *Gâteaux differentiable* at $x_0 \in X$.

Remark: *Every Fréchet differentiable function is Gâteaux differentiable, but the converse is not true.*

Theorem: *Let X and P be Banach spaces. $L : X \longrightarrow P$ is Lipschitz and $\dim(X) < \infty$ then concepts of Gâteaux differentiability and Fréchet differentiability coincide.*

Appendix C

Minimum of Symmetric Forms

If the form is symmetric the evaluation can be better:

- If a is symmetric v is solution of $a(u,v) = L(u) \quad \forall u \in H \Leftrightarrow v = \min_{u \in H} J(u) = \frac{1}{2}a(u,u) - L(u)$.
- By Galerkin and Lax-Milgram v_m is minimum of $J(u)$ in H_m.
- v_m is also minimum of $2J(u)+a(v,v)$ in H_m, where v is the solution of the system.
- In H_m we have:

$$2J(v_m) + a(v,v) = a(v_m, v_m) - 2a(v, v_m) + a(v,v) =$$

$$= a(v - v_m, v - v_m) \leq a(v - w, v - w) \quad \forall w \in H_m.$$

- By limitation and coercitivity of the form:

$$\|v - v_m\|_H^2 \leq \frac{1}{c_2} a(v - w, v - w) \leq$$

$$\leq \frac{c_1}{c_2} \|v - w\|_H^2 \quad \forall w \in H_m.$$

- By squaring and searching minimum in H_m we obtain the following majoration of the error:

$$\|v - v_m\|_H \leq \sqrt{\frac{c_1}{c_2}} \min_{u \in H_m} \|v - m\|_H .$$

Appendix D

The Gronwall's Lemma

To evaluate $\sup_{t\in[0,2\pi]} |\dot{v}(t)| = K$, $\sup_{t\in[0,2\pi]} |\ddot{v}(t)| = T$ and $\sup_{t\in[0,2\pi]} |\dddot{v}(t)| = Q$ we can apply Gronwall:

Gronwall's Lemma: We assume $f, y, c : I \longrightarrow R^+$ continuous functions, $c > 0$ a constant and $x_0 \in I$. If $\forall x \in I$:

$$y(t) \leq \left| \int_{x_0}^{x} f(t)y(t)dt \right| + c(t) \tag{D.1}$$

then, $\forall x \in I$ holds:

$$0 \leq y(t) \leq c(t)e^{|\int_{x_0}^{x} f(t)y(t)dt|}. \tag{D.2}$$

Integrating equation 6.3 we obtain (considering for simplicity unitary mass):

$$\dot{v} = \int_0^t f(\tau)d\tau - \xi \int_0^t \dot{v}d\tau + \int_0^t g(\tau)vd\tau \tag{D.3}$$

where $g(t) = 2k_1(x(t) - \lambda_1)(t) + 2k_2(l - x(t) - \lambda_2(t))$.

We know that $f(t)$ and $g(t)$ are C^1 functions in I and for Weierstrass they (and their derivatives) have a maximum and a minimum in I:

$$\alpha = \max_I |g(\tau)| \quad ; \quad \beta = \max_I |f(\tau)|;$$

$$\gamma = \max_I |\dot{g}(\tau)| \quad ; \quad \delta = \max_I |\dot{f}(\tau)|.$$

We obtain:

$$|\dot{v}| \leq \beta|t| + \alpha \left| \int_0^t v(\tau)d\tau \right| + \xi \left| \int_0^t \dot{v}(\tau)d\tau \right| =$$

$$= \beta|t| + \alpha \left| tv(t) - \int_0^t \dot{v}(\tau)d\tau \right| + \xi \left| \int_0^t \dot{v}(\tau)d\tau \right| \leq$$

$$\leq \beta|t| + \alpha|t-1| \left| \int_0^t \dot{v}(\tau)d\tau \right| + \xi \left| \int_0^t \dot{v}(\tau)d\tau \right|.$$

Imposing $M = \max_I |\xi + \alpha|t-1||$ we have:

$$|\dot{v}| \leq M \left| \int_0^t \dot{v}d\tau \right| + \beta|t|. \tag{D.4}$$

The hypothesis of *Gronwall's Lemma* are satisfied, applying it we obtain the evaluation for $K = \sup_I |\dot{v}|$:

$$0 \leq |\dot{v}(t)| \leq \beta|t|e^{tM}. \tag{D.5}$$

Using stime D.5 we can evaluate $T = \sup_I |\ddot{v}|$ and $Q = \sup_I |\ \dddot{v}\ |$. From equation 6.3 we obtain:

$$|\ddot{v}| \leq \xi|\dot{v}| + \alpha \left| \int_0^t \dot{v}d\tau \right| + \beta. \tag{D.6}$$

By diseq.D.5 we obtain:

$$|\ddot{v}| \leq \xi\beta|t|e^{tM} + \alpha \left| \int_0^t \tau\beta e^{\tau M} d\tau \right| + \beta.$$

Solving the simple integral of the second member of disequation above we obtain:

$$|\ddot{v}| \leq \xi\beta|t|e^{tM} + \frac{\alpha\beta}{M^2}\left(e^{tM}|tM-1|+1\right) + \beta.$$

Grouping we obtain the evaluation for $T = \sup_I |\ddot{v}|$:

$$0 \leq |\ddot{v}(t)| \leq \beta e^{tM} \left(\xi |t| + \alpha \frac{|tM - 1|}{M^2} \right) + \beta \left(\frac{\alpha}{M^2} + 1 \right). \tag{D.7}$$

To evaluate $\sup_I | \dddot{v} |$ we can derive eq. 6.3. We obtain:

$$\dddot{v}\ (t) = -\xi \ddot{v}(t) + \dot{g}(t) v(t) + g(t) \dot{v}(t) + \dot{f}(t). \tag{D.8}$$

From equation D.8 we obtain:

$$| \dddot{v}\ (t)| \leq \xi |\ddot{v}(t)| + \gamma \left| \int_0^t \dot{v} d\tau \right| + \alpha |\dot{v}(t)| + \delta.$$

Solving the integral and by diseq.D.5 and D.7 we have:

$$| \dddot{v} | \leq \xi \beta e^{tM} \left(\xi \beta |t| + \alpha \frac{|tM - 1|}{M^2} \right) + \xi \beta \left(\frac{\alpha}{M^2} + 1 \right) +$$

$$+ \frac{\gamma \beta}{M^2} \left(e^{tM} |tM - 1| + 1 \right) + \alpha \beta |t| e^{tM} + \delta.$$

Grouping we obtain the evaluation for $Q = \sup_I | \dddot{v} |$:

$$0 \leq |\dddot{v}| \leq \beta e^{tM} \left[(\xi^2 + \alpha) |t| + (\alpha \xi + \gamma) \frac{|tM - 1|}{M^2} \right] + \beta \frac{(\alpha \xi + \gamma)}{M^2} + \xi \beta + \delta. \tag{D.9}$$

Appendix E

Simulations with MAPLE

E.1 Example of Monodimensional System

Evaluation of stiffness for the system illustrated in Fig.3.2.

```
> restart; with(linalg):
> eq2:=-S+m*diff(diff(v(t),t),t)+nu*diff(v(t),t)+
  2*k1*(x(t)-lambda1(t))*v(t)+2*k2*(l-x(t)-
  lambda2(t))*v(t):
> eq1:=L-m*diff(diff(x(t),t),t)+nu*diff(x(t),t)-k1*
  (x(t)-lambda1(t))^2+k2*(l-x(t)-lambda2(t))^2:
> L:=4:m:=1:l:=10:nu:=0:k2:=1:k1:=1:x(t):=5:
> lam1(t):=3-t:
> lam2(t):=solve(subs(lambda1(t)=lam1(t),
  eq1),lambda2(t))[1]:
> plot([lam1(t),lam2(t)],t=0..2*Pi,colour=[red,blue]);
> vs(t):=sin(k*t)/sqrt(Pi*(1+k^2+k^4)):
> vc(t):=cos(k*t)/sqrt(Pi*(1+k^2+k^4)):
> S1c(t):=S+subs(lambda2(t)=lam2(t),lambda1(t)=lam1(t),
```

```
  v(t)=vc(t),eq2:
> S1s(t):=S+subs(lambda2(t)=lam2(t),
  lambda1(t)=lam1(t),v(t)=vs(t),eq2):
> Stiff1c:=simplify(eval(S1c(t))):
  Stiff1s:=simplify(eval(S1s(t))):
> lam11(t):=1-2*t:lam21(t):=solve(subs
  (lambda1(t)=lam11(t),eq1),lambda2(t))[1]:
> plot([lam11(t),lam21(t)],t=0..2*Pi,colour=[red,blue]):
> S2c(t):=S+subs(lambda2(t)=lam21(t),lambda1(t)=
  lam11(t),v(t)=vc(t),eq):S2s(t):=S+subs(lambda2(t)=
  lam21(t),lambda1(t)=lam11(t),v(t)=vs(t),eq):
> Stiff2c:=simplify(eval(S2c(t))): Stiff2s:=
  simplify(eval(S2s(t))):
> plot([subs(k=1,Stiff1s*sqrt(Pi)),subs(k=1,Stiff2s)*
  sqrt(Pi)],t=0..2*Pi,colour=[red,blue]):
```

Evaluation of stiffness for the system illustrated in Fig.3.2 with piecewise constant controls.

```
> restart; with(linalg):
> eq2:=-S+m*diff(diff(v(t),t),t)+nu*diff(v(t),t)+
  2*k1*(x(t)-lambda1(t))*v(t)+2*k2*(l-x(t)-
  lambda2(t))*v(t):
> eq1:=L-m*diff(diff(x(t),t),t)-nu*diff(x(t),t)-k1*
  (x(t)-lambda1(t))^2+k2*(l-x(t)-lambda2(t))^2:
> L:=4:m:=1:l:=10:nu:=1:k2:=1:k1:=1:eq1:
> lambda1(t):=piecewise(t<2, 3, t>=2, 5):
> lambda2(t):=piecewise(t<2, 5, t>=2, 6):
> plot([lambda1(t),lambda2(t)],t=0..5,
  colour=[blue,red]):
> g(t):=simplify(dsolve({eq1,D(x)(0)=5,x(0)=0},x(t))):
```

```
> g1(t):=convert(rhs(g(t)),list)[2]:
> g2(t):=convert(rhs(g(t)),list)[4]:
> h:=piecewise(t<2, g1(t), t>=2, g2(t)):
> plot(h(t),t=0..4):
> S(t):=solve(eq2,S):
> S1(t):=convert(S(t),list)[2]:
> S2(t):=convert(S(t),list)[4]:
> St:=piecewise(t<2, S1(t), t>=2, S2(t)):
> eval(subs(v(t)=1/sqrt(Pi*(k^4+k^2+1))*
  sin(k*t),S1(t))):
> eval(subs(v(t)=1/sqrt(Pi*(k^4+k^2+1))*
  cos(k*t),S1(t))):
> eval(subs(v(t)=1/sqrt(Pi*(k^4+k^2+1))*
  sin(k*t),S2(t))):
> eval(subs(v(t)=1/sqrt(Pi*(k^4+k^2+1))*
  cos(k*t),S2(t))):
> n:=10: for i from 1 to 2*n+1 do if (i=1)
  then v(i):=1/sqrt(2*Pi) elif (i mod 2=0)
  then v(i):=1/sqrt(Pi)*sin(i/2*t) else
  v(i):=1/sqrt(Pi)*cos(((i-1)/2)*t) end if:od:
> fc:= proc(n) local a:
  a := array( [ seq (v(i),i=1..n)]):
> return eval( a, 1 ) end proc:base1:=fc(2*n+1):
> for i from 1 to 2*n+1 do if (i=1) then
  v(i):=1/sqrt(2*Pi) elif (i mod 2=0) then
  v(i):=1/sqrt(Pi*((i/2)^4+(i/2)^2+1))*
  sin(i/2*t) else v(i):=1/sqrt(Pi*(((i-1)/2)^4+
 ((i-1)/2)^2+1))*cos(((i-1)/2)*t) end if: od:
> fc:= proc(n) local a:
```

```
  a := array([seq (v(i),i=1..n)]):
> return eval( a, 1 ) end proc:base2:=fc(2*n+1):
> fc:= proc(n) local a:
  a := array( [seq(eval(simplify(solve(subs(v(t)=
  base2[i],eq2),S))),i=1..n) ]):
  return eval( a, 1 ) end proc:Stiff:=fc(2*n+1):
> v[1]:=base2[1]:
> Stiff11(t):=convert(Stiff[1],list)[2]:
> Stiff12(t):=convert(Stiff[1],list)[4]:
> Sti1:=piecewise(t<2, Stiff11(t),t>=2,Stiff12(t)):
> v[2]:=base2[2]:
> Stiff21(t):=convert(Stiff[2],list)[2]:
> Stiff22(t):=convert(Stiff[2],list)[4]:
> Sti2:=piecewise(t<2, Stiff21(t),t>=2,Stiff22(t)):
> v[3]:=base2[3]:
> Stiff31(t):=convert(Stiff[3],list)[2]:
> Stiff32(t):=convert(Stiff[3],list)[4]:
> Sti3:=piecewise(t<2,Stiff31(t),t>=2,Stiff32(t)):
> v[4]:=base2[4]:
> Stiff41(t):=convert(Stiff[4],list)[2]:
> Stiff42(t):=convert(Stiff[4],list)[4]:
> Sti4:=piecewise(t<2, Stiff41(t),t>=2,Stiff42(t)):
> v[11]:=base2[11]:
> Stiff111(t):=convert(Stiff[11],list)[2]:
> Stiff112(t):=convert(Stiff[11],list)[4]:
> Sti11:=piecewise(t<2,Stiff111(t),t>=
  2,Stiff112(t)):
> v[21]:=base2[21]:
> Stiff211(t):=convert(Stiff[21],list)[2]:
```

```
> Stiff212(t):=convert(Stiff[21],list)[4]:
> Sti21:=piecewise(t<2,Stiff211(t),t>=
  2,Stiff212(t)):
> plot([Sti1,Sti2,Sti3,Sti4,Sti11,Sti21],t=0..5,
  colour=[yellow,red,green,gray,blue,black]):
```

E.2 Example of Bidimensional System

Evaluation of stiffness for the system illustrated in Fig.5.1.

```
> restart; with(linalg):
> l1:=2:l2:=1:r:=1:m1:=2:m2:=1:g:=9.81:nu1:=0:nu2:=0:
  k1:=2:k2:=1:k3:=1: theta1:=eta1(t):theta2:=eta2(t):
> I1:=eval(1/12*m1*l1^2+(l1/2)^2*m1): I2:=eval(1/12*
  m2*l2^2+d*m2):
> d:=eval(l1^2+(l2/2)^2-2*l1*(l2/2)*cos(theta2)):
> x1:=eval(l1+l2/2+r*theta1+r*theta2):
> x2:=eval(l1+l2/2-r*theta1-r*theta2):
> x3:=eval(l1/2-r*theta1):
> F1:=vector(2,[k1*(x1-lambda1)^2*sin(theta1+theta2),
  k1*(x1-lambda1)^2*cos(theta1+theta2)]):
> F2:=vector(2,[k2*(x2-lambda2)^2*sin(theta1+theta2),
  k2*(x2-lambda2)^2*cos(theta1+theta2)]):
> F3:=vector([k3*(x3-lambda3)^2*sin(theta1),k3*(x3-
  lambda3)^2*cos(theta1)]):
> C:=<Cx,Cy,0>:
> bCx:= l1*cos(theta1) + l2*cos(theta1 + theta2):
> bCy:= l1*sin(theta1) + l2*sin(theta1 + theta2):
> bF1:=vector([l1*cos(theta1)+l2/2*cos(theta1+theta2),
```

```
  l1*sin(theta1)+l2/2*sin(theta1+theta2)]):bF2:=bF1:
> bF3:=vector([l1/2*cos(theta1),l1/2*sin(theta1)]):
> P11:=m1*g*1/2*l1*cos(theta1):
> P21:=m2*g*(l1*cos(theta1)+1/2*l2*cos(theta1+theta2)):
> with(VectorCalculus):
> F11:=simplify(CrossProduct(<F1[1],F1[2],0>,<bF1[1],
  bF1[2],0>))[3]:
> F21:=simplify(CrossProduct(<F2[1],F2[2],0>,
  <bF2[1],bF2[2],0>))[3]:
> F31:=simplify(CrossProduct(<F3[1],F3[2],0>,
  <bF3[1],bF3[2],0>))[3]:
> C1:=simplify(CrossProduct(C,<bCx,bCy,0>))[3]:
> xi1:=eval(F11+F21+F31+C1+P11+P21+diff(theta1,t)*nu1):
> F12:=simplify(k1*(x1-lambda1)^2*l2/2):
> F22:=simplify(k2*(x2-lambda2)^2*l2/2):
> C2:=Cy*l2*cos(theta1+theta2)-Cx*l2*cos(theta1+theta2):
> P22:=m2*g*1/2*l2*cos(theta1+theta2):
> xi2:=eval(F12+F22+P22+C2+diff(theta2,t)*nu2):
> eq1:=eval(expand((I1+m1*l1^2+m2*((l1/2)^2+l2^2+2*l2*
  l1/2*cos(theta2)))*diff(diff(theta1,t),t)+(I2+m2*
  (l2^2+l2*l1/2*cos(theta2)))*diff(diff(theta2,t),
  t)-2*m2*l2*l1/2*sin(theta2)*diff(theta1,t)*
  diff(theta2,t)-m2*l2*l1/2*sin(theta2)*
  diff(theta2,t)^2+(m1*l1+m2*l1/2)*g*cos(theta1)+
  m2*l2*g*cos(theta1+theta2)-xi1,trig)):
> eq2:=eval((I2+m2*(l2^2+l2*l1/2*cos(theta2)))*
  diff(diff(theta1,t),t)+(I2+m2*l2)*
  diff(diff(theta2,t),t)+m2*l2*l1/2*sin(theta2)*
  diff(theta1,t)^2+m2*l2*g*cos(theta1+theta2)-xi2):
```

```
> sol:=solve({eq1,eq2},{Cx,Cy}):
> CY:=rhs(sol[1]): CX:=rhs(sol[2]):
> CXs1:=subs(theta1=theta1+h1*v1(t),CX):
> CXs2:=subs(theta2=theta2+h2*v2(t),CX):
> CYs1:=subs(theta1=theta1+h1*v1(t),CY):
> CYs2:=subs(theta1=theta1+h2*v2(t),CY):
> rapinc11:=simplify((CXs1-CX)/h1):
> rapinc12:=simplify((CXs2-CX)/h2):
> rapinc21:=simplify((CYs1-CY)/h1):
> rapinc22:=simplify((CYs2-CY)/h2):
> with(Student[VectorCalculus]):
> DiffGat11:=limit(rapinc11,h1=0):
> DiffGat12:=limit(rapinc12,h2=0):
> DiffGat21:=limit(rapinc21,h1=0):
> DiffGat22:=limit(rapinc22,h2=0):
> Stiffx:=DiffGat11+DiffGat12:
> Stiffy:=DiffGat21+DiffGat22:
> lambda2:=5-t:
> lamb3:=solve(eq1c,lambda3)[1]:
> eq1s:=subs(Cy=-20,subs(Cx=0,eq1)):
> eq2s:=subs(Cy=-20,subs(Cx=0,eq2)):
> eq1c:=eval(subs({theta1=0,theta2=Pi/3},eq1s)):
> eq2c:=eval(subs({theta1=0,theta2=Pi/3},eq2s)):
> eq1d:=subs(lambda3=lamb3,eq1c):
> lam1:=solve(eq1d,lambda1):
> lam3:=evalf(simplify(subs(lambda1=lam1,lamb3))):
> plot(4.8+2.5*t-0.34*t^2,t=0..9):
> Sx:=simplify(eval(subs({lambda1=lam1,
  lambda3=lam3,theta1=0,theta2=Pi/3,v1(t)=sin(t),
```

```
  v2(t)=cos(t)},Stiffx))):
> Sy:=simplify(eval(subs({lambda1=lam1,
  lambda3=lam3,theta1=0,theta2=Pi/3,v1(t)=sin(t),
  v2(t)=cos(t)},Stiffy))):
> Stiffness:=<Sx,Sy>:
> plot([Sx,Sy], t=0..8, colour=[blue,red]):
```

E.3 Example of Threedimensional System

Evaluation of stiffness for the system illustrated in Fig.5.5.

```
> restart; with(LinearAlgebra):
> a[1]:=10:a[2]:=10:a[3]:=10:l[1]:=a[1]/2:
> l[2]:=a[2]/2:l[3]:=a[3]/2:r[1]:=1:r[2]:=1:r[3]:=1:
> m[l1]:=5:m[l2]:=5:m[l3]:=5:
> w:=Vector([xi[1](t),xi[2](t),xi[3](t)]):
> k[1]:=1:k[2]:=1:k[3]:=1:k[4]:=1:k[5]:=1:k[6]:=1:
> tau[1](t):=k[1]*(theta[1](t)-lambda[1])^2-k[2]*
  (theta[1](t)-lambda[2])^2:tau[2](t):=k[3]*
  (theta[2](t)-lambda[3])^2-k[4]*(theta[2](t)-
  lambda[4])^2:
> tau[3](t):=k[5]*(theta[3](t)-lambda[5])^2-k[6]*
  (theta[3](t)-lambda[6])^2:
> Tau:=<tau[1](t),tau[2](t),tau[3](t)>:
> h:=1:f[v1]:=1:f[v2]:=1:f[v3]:=1:f[s1]:=1:f[s2]:=1:
> f[s3]:=1:nu[1]:=1:nu[2]:=1:nu[3]:=1:m[m1]:=5:
> m[m2]:=5:m[m3]:=5:kr[r1]:=1:kr[r2]:=1:kr[r3]:=1:
> c[1]:=cos(theta[1](t)):s[1]:=sin(theta[1](t)):
> c[2]:=cos(theta[2](t)):s[2]:=sin(theta[2](t)):
```

```
> c[3]:=cos(theta[3](t)):s[3]:=sin(theta[3](t)):
> c[11]:=cos(theta[1](t)+theta[1](t)):
> c[12]:=cos(theta[1](t)+theta[2](t)):
> c[13]:=cos(theta[1](t)+theta[3](t)):
> c[21]:=cos(theta[2](t)+theta[1](t)):
> c[22]:=cos(theta[2](t)+theta[2](t)):
> c[23]:=cos(theta[2](t)+theta[3](t)):
> c[31]:=cos(theta[3](t)+theta[1](t)):
> c[32]:=cos(theta[3](t)+theta[2](t)):
> c[33]:=cos(theta[3](t)+theta[3](t)):
> s[11]:=sin(theta[1](t)+theta[1](t)):
> s[12]:=sin(theta[1](t)+theta[2](t)):
> s[13]:=sin(theta[1](t)+theta[3](t)):
> s[21]:=sin(theta[2](t)+theta[1](t)):
> s[22]:=sin(theta[2](t)+theta[2](t)):
> s[23]:=sin(theta[2](t)+theta[3](t)):
> s[31]:=sin(theta[3](t)+theta[1](t)):
> s[32]:=sin(theta[3](t)+theta[2](t)):
> s[33]:=sin(theta[3](t)+theta[3](t)):
> q:=Vector([theta[1](t),theta[2](t),theta[3](t)]):
> Dq:=Vector([diff(theta[1](t),t),diff(theta[2](t),t)
  diff(theta[3](t),t)]):
> DDq:=Vector([diff(diff(theta[1](t),t),t),diff(diff
  (theta[2](t),t),t),diff(diff(theta[3](t),t),t)]):
> A[01]:=Matrix([[c[1],0,s[1],0],[s[1],0,-c[1],0],
  [0,-1,0,0],[0,0,0,1]]):
> A[12]:=Matrix([[c[2],-s[2],0,a[2]*c[2]],[s[2],
  c[2],0,a[2]*s[2]],[0,0,1,0],[0,0,0,1]]):
> A[23]:=Matrix([[c[3],s[3],0,a[3]*c[3]],[s[3],
```

```
  c[3],0,a[3]*s[3]],[0,0,1,0],[0,0,0,1]]):
> T[01]:=A[01]:T[02]:=Multiply(A[01],A[12]):
> T[03]:=Multiply(T[02],A[23]):
> MI[l1]:=Matrix([[im[l1x],0,0],[0,im[l1y],0],
  [0,0,im[l1z]]]):
> MI[l2]:=Matrix([[im[l2x],0,0],[0,im[l2y],0],
  [0,0,im[l2z]]]):
> MI[l3]:=Matrix([[im[l3x],0,0],[0,im[l3y],0],
  [0,0,im[l3z]]]):
> MI[m1]:=Matrix([[im[m1x],0,0],[0,im[m1y],0],
  [0,0,im[m1z]]]):
> MI[m2]:=Matrix([[im[m2x],0,0],[0,im[m2y],0],
  [0,0,im[m2z]]]):
> MI[m3]:=Matrix([[im[m3x],0,0],[0,im[m3y],0],
  [0,0,im[m3z]]]):
> im[l1x]:=1:im[l1y]:=1:im[l1z]:=1:im[l2x]:=1:
> im[l2y]:=1:im[l2z]:=1:im[l3x]:=1:im[l3y]:=1:
> im[l3z]:=1:im[m1x]:=1:im[m1y]:=1:im[m1z]:=1:
> im[m2x]:=1:im[m2y]:=1:im[m2z]:=1:im[m3x]:=1:
> im[m3y]:=1:im[m3z]:=1:
> Fv:=Matrix([[f[v1],0,0],[0,f[v2],0],[0,0,f[v3]]]):
> fs:=Vector([f[s1]*sign(diff(theta[1](t),
  t)),f[s2]*sign(diff(theta[2](t),t)),f[s3]*
  sign(diff(theta[3](t),t))]):
> p[0]:=<0,0,0>:p[1]:=<0,0,0>:p[l1]:=<0,0,0>:
> p[2]:=<a[2]*c[1]*c[2],a[2]*s[1]*c[2],a[2]*s[2]>:
> p[l2]:=<l[2]*c[1]*c[2],l[2]*s[1]*c[2],l[2]*s[2]>:
> p[3]:=<c[1]*(a[2]*c[2]+a[3]*c[23]),s[1]*
  (a[2]*c[2]+a[3]*c[23]),a[2]*s[2]+a[3]*s[23]>:
```

```
> p[l3]:=<c[1]*(a[2]*c[2]+l[3]*c[23]),s[1]*
  (a[2]*c[2]+l[3]*c[23]),a[2]*s[2]+l[3]*s[23]>:
> z[0]:=<0,0,1>:z[1]:=T[01][1..3,3]:z[2]:=T[02][1..3,3]:
> JP[l1]:=Matrix([CrossProduct(z[0],(p[l1]-p[0])),
  p[0],p[0]]):JO[l1]:=Matrix([z[0],p[0],p[0]]):
> JP[m1]:=Matrix([CrossProduct(z[0],(p[1]-p[0])),
  p[0],p[0]]):
> JO[m1]:=Matrix([Multiply(z[0],kr[r1]),p[0],p[0]]):
> JP[l2]:=Matrix([CrossProduct(z[0],(p[l2]-p[0])),
  CrossProduct(z[1],(p[l2]-p[1])),p[0]]):
> JO[l2]:=Matrix([z[0],z[1],p[0]]):
> JP[m2]:=Matrix([CrossProduct(z[0],(p[2]-p[0])),
  CrossProduct(z[1],(p[2]-p[1])),p[0]]:
> JO[m2]:=Matrix([JO[l2][1..3,1],
  Multiply(z[1],kr[r2]),p[0]]):
> JP[l3]:=Matrix([CrossProduct(z[0],(p[l3]-p[0])),
  CrossProduct(z[1],(p[l3]-p[1])),
  CrossProduct(z[2],(p[l3]-p[2]))]):
> JP[m3]:=Matrix([CrossProduct(z[0],(p[3]-p[0])),
  CrossProduct(z[1],(p[3]-p[1])),
  CrossProduct(z[2],(p[3]-p[2]))]):
> JO[l3]:=Matrix([z[0],z[1],z[2]]):
> JO[m3]:=Matrix([JO[l3][1..3,1],
  JO[l3][1..3,2],Multiply(z[2],kr[r3])]):
> J:=Matrix([[-s[1]*(a[2]*c[2]+a[3]*c[23]),
  -c[1]*(a[2]*s[2]+a[3]*s[23]),-a[3]*c[1]*s[23]],
  [c[1]*(a[2]*c[2]+a[3]*c[23]),-s[1]*(a[2]*s[2]+
  a[3]*s[23]),-a[3]*
  s[1]*s[23]],[0,a[2]*c[2]+a[3]*c[23],a[3]*c[23]]]):
```

```
> fg:=9.81:g[0]:=<0,fg,0>:
  g[1]:=Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[l1][1..3,1])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[m1][1..3,1])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[l2][1..3,1])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[m2][1..3,1])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[l3][1..3,1])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[m3][1..3,1]):
> g[2]:=Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[l1][1..3,2])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[m1][1..3,2])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[l2][1..3,2])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[m2][1..3,2])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[l3][1..3,2])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[m3][1..3,2]):
> g[3]:=Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[l1][1..3,3])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l1]),JP[m1][1..3,3])+
```

```
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[l2][1..3,3])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l2]),JP[m2][1..3,3])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[l3][1..3,3])+
  Multiply(ScalarMultiply(Transpose(g[0]),
  m[l3]),JP[m3][1..3,3]):G:=<g[1],g[2],g[3]>:
> B1:=ScalarMultiply(Multiply(Transpose(JP[l1]),
JP[l1]),
  m[l1])+Multiply(Multiply(Transpose(JO[l1]),
  MI[l1]),JO[l1])+
  ScalarMultiply(Multiply(Transpose(JP[m1]),JP[m1]),
  m[m1])+Multiply(Multiply(Transpose(JO[m1]),MI[m1]),
  JO[m1])+ScalarMultiply(Multiply(Transpose(JP[l2]),
  JP[l2]),m[l2])+Multiply(Multiply(Transpose(JO[l2]),
  MI[l2]),JO[l2])+
  ScalarMultiply(Multiply(Transpose(JP[m2]),
  JP[m2]),m[m2])+Multiply(Multiply(Transpose(JO[m2]),
  MI[m2]),JO[m2])+
  ScalarMultiply(Multiply(Transpose(JP[l3]),JP[l3]),
  m[l3])+Multiply(Multiply(Transpose(JO[l3]),MI[l3]),
  JO[l3])+ ScalarMultiply(Multiply(Transpose(JP[m3]),
  JP[m3]),m[m3])+Multiply(Multiply(Transpose(JO[m3]),
  MI[m3]),JO[m3]):
> B:=combine(%,trig):
> for i from 1 to 3 do for j from 1 to 3 do
  a[i,j]:=i+j od:od:
> for i from 1 to 3 do for j from 1 to 3 do
```

```
  for k from 1 to 3 do
  ch[i,j,k]:=eval(1/2*(diff(subs({theta[1](t)=theta[1],
  theta[2](t)=theta[2],theta[3](t)=theta[3]},B[i,j]),
  theta[k])+diff(subs({theta[1](t)=theta[1],theta[2](t)=
  theta[2],theta[3](t)=theta[3]},B[i,k]),theta[j])-
  diff(subs({theta[1](t)=theta[1],theta[2](t)=theta[2],
  theta[3](t)=theta[3]},B[j,k]),theta[i]))) od :od: od:
> Chr:=subs({theta[1]=theta[1](t),theta[2]=theta[2](t),
  theta[3]=theta[3](t)},array([seq(seq(seq(1/2*
  (diff(subs({theta[1](t)=theta[1],theta[2](t)=theta[2],
  theta[3](t)=theta[3]},B[i,j]),theta[k])+
  diff(subs({theta[1](t)=theta[1],theta[2](t)=
  theta[2],theta[3](t)=theta[3]},B[i,k]),theta[j])-
  diff(subs({theta[1](t)=theta[1],theta[2](t)=theta[2],
  theta[3](t)=theta[3]},B[j,k]),theta[i])), k=1..3),
  j=1..3),i=1..3 )])):
> tot :=0: for i from 1 to 3 do tot := Chr[i]*
  Dq[i]+tot end do:
> ch[11]:=tot:eval(%):tot :=0: for i from 1 to 3 do
  tot := Chr[i+3]*Dq[i]+tot end do: ch[12]:=eval(tot):
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+6]*Dq[i]+tot end do:  ch[13]:=tot:
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+9]*Dq[i]+tot end do: ch[21]:=tot:
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+12]*Dq[i]+tot end do: ch[22]:=tot:
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+15]*Dq[i]+tot end do: ch[23]:=tot:
  tot :=0: for i from 1 to 3 do
```

```
  tot := Chr[i+18]*Dq[i]+tot end do: ch[31]:=tot:
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+21]*Dq[i]+tot end do: ch[32]:=tot:
  tot :=0: for i from 1 to 3 do
  tot := Chr[i+24]*Dq[i]+tot end do: ch[33]:=tot:
> C:=Matrix([[ch[11],ch[12],ch[13]],[ch[21],ch[22],
  ch[23]],[ch[31],ch[32],ch[33]]]):
  BD:=Matrix(3,3):for i from 1 to 3 do
  for j from 1 to 3 do
  BD[i,j]:=diff(B[i,j],t) od:od:BD:combine(BD-2*C,trig):
> eq:=Multiply(B,DDq)+Multiply(C,Dq)+Multiply(Fv,Dq)+
  G-Tau+Multiply(Transpose(J),w):
> Ok:=Transpose(-Tau+Multiply(B,DDq)+Multiply(C,Dq)+
  Multiply(Fv,Dq)+G):
> IJ:=simplify(MatrixInverse(J)):
> wrt:=simplify(Multiply(Ok,Transpose(IJ))):
> wr:=(subs({theta[1](t)=theta[1],theta[2](t)=
  theta[2],theta[3](t)=theta[3]},wrt):
> wr1:=wr[1]:wr2:=wr[2]:wr3:=wr[3]:
> wr1s:=subs(theta[1]=theta[1]+h1*v1(t),wr1):
> wr1s2:=subs(theta[2]=theta[2]+h2*v2(t),wr1):
> wr1s3:=subs(theta[3]=theta[3]+h3*v3(t),wr1):
> wr2s:=subs(theta[1]=theta[1]+h1*v1(t),wr2):
> wr2s2:=subs(theta[2]=theta[2]+h2*v2(t),wr2):
> wr2s3:=subs(theta[3]=theta[3]+h3*v3(t),wr2):
> wr3s:=subs(theta[1]=theta[1]+h1*v1(t),wr3):
> wr3s2:=subs(theta[2]=theta[2]+h2*v2(t),wr3):
> wr3s3:=subs(theta[3]=theta[3]+h3*v3(t),wr3):
> rapinc11:=simplify((wr1s-wr1)/h1):
```

```
> rapinc12:=simplify((wr1s2-wr1)/h2):
> rapinc13:=simplify((wr1s3-wr1)/h3):
> rapinc21:=simplify((wr2s-wr2)/h1):
> rapinc22:=simplify((wr2s2-wr2)/h2):
> rapinc23:=simplify((wr2s3-wr2)/h3):
> rapinc31:=simplify((wr3s-wr3)/h1):
> rapinc32:=simplify((wr3s2-wr3)/h2):
> rapinc33:=simplify((wr3s3-wr3)/h3):
> with(Student[VectorCalculus]):
> DiffGat11:=limit(rapinc11,h1=0):
> DiffGat12:=limit(rapinc12,h2=0):
> DiffGat13:=limit(rapinc13,h3=0):
> DiffGat21:=limit(rapinc21,h1=0):
> DiffGat22:=limit(rapinc22,h2=0):
> DiffGat23:=limit(rapinc23,h3=0):
> DiffGat31:=limit(rapinc31,h1=0):
> DiffGat32:=limit(rapinc32,h2=0):
> DiffGat33:=limit(rapinc33,h3=0):
> Stiffx:=DiffGat11+DiffGat12+DiffGat13:
> Stiffy:=DiffGat21+DiffGat22+DiffGat23:
> Stiffz:=DiffGat31+DiffGat32+DiffGat33:
> lambda[2]:=2-t:lambda[4]:=1-1/2*t:lambda[6]:=-t:
> eqs:=subs(xi[1]=10,xi[2]=0,xi[3]=50,
  theta[1](t)=theta[1],theta[2](t)=theta[2],
  theta[3](t)=theta[3],eq):
> theta[1]:=-Pi/4:theta[2]:=Pi/4:theta[3]:=-Pi/2:
> lambd1:=solve(eqs[1],lambda[1]):
> lamb3:=solve(subs(lambda[1]=lambd1,eqs[2]),lambda[3]):
> lam5:=solve(subs({lambda[1]=lambd1,lambda[3]=lamb3},
```

```
  eqs[3]),lambda[5])[1]:lam3:=lamb3[1]:lam1:=lambd1[1]:
> plot(16*t^2-50.3*t+610,t=0..20):
> plot(4*t^2-3.4*t+19162,t=0..20):
> plot(16*t^2-89.13*t+21344.8,t=0..20):
> v1(t):=sin(t):v2(t):=sin(2*t):v3(t):=cos(t):
> Sx:=simplify(evalf(subs({lambda[1]=lam1,
  lambda[3]=lam3,lambda[5]=lam5,
  theta[1](t)=-Pi/4,theta[2](t)=Pi/4,
  theta[3](t)=-Pi/2},Stiffx))):
> Sy:=simplify(evalf(subs({lambda[1]=lam1,
  lambda[3]=lam3,lambda[5]=lam5,theta[1](t)=-Pi/4,
  theta[2](t)=Pi/4,theta[3](t)=-Pi/2},Stiffy))):
> Sz:=simplify(evalf(subs({lambda[1]=lam1,
  lambda[3]=lam3,lambda[5]=lam5,
  theta[1](t)=-Pi/4,theta[2](t)=Pi/4,
  theta[3](t)=-Pi/2},Stiffz))):
> plot([Sx,Sy,Sz], t=0..16, colour=[blue,red,green]):
```

E.4 Orthonormalization of the Basis

Building of an orthonormal basis for space H (def.6.6) using Grahm-Schmidt Orthonormalization Procedure (fig. 6.3).

```
> restart;with(linalg):
> phi(t):=-cos(8*t)+1;plot(phi(t),t=0..Pi/4);
> n:=7;
> for i from 1 to n do v(i):=piecewise(t>=(i-1)*
  Pi/(4*(n+1)) and t<(i+1)*Pi/(4*(n+1)),
```

```
  -cos(4*(n+1)*(t-(i-1)*Pi/(4*(n+1))))+1):od:
> fc:= proc(n) local a;
  a := array( [ seq (v(i),i=1..n) ]);
  return eval( a, 1 ) end proc:base:=fc(n):
> plot(base,t=0..Pi/4);
> for i from 1 to n do dv(i):=diff(v(i),t):od:
> fc:= proc(n) local a;
  a := array( [ seq (dv(i),i=1..n) ]);
  return eval( a, 1 ) end proc:dbase:=fc(n):
> plot(dbase,t=0..Pi/4);
> for i from 1 to n do
  ddv(i):=diff(diff(v(i),t),t):od:
> fc:= proc(n) local a;
  a := array( [ seq (ddv(i),i=1..n) ]);
  return eval( a, 1 ) end proc:ddbase:=fc(n):
> plot(ddbase,t=0..Pi/4);
> f:=v->sqrt(int(v^2+diff(v,t)^2+
  diff(diff(v,t),t)^2,t=0..Pi/4));
> g:=(v,w)->int(v*w+diff(v,t)*diff(w,t)+
  diff(diff(v,t),t)*diff(diff(w,t),t),t=0..Pi/4);
> for i from 1 to n do if i=1 then v[i]:=base[1] else
  v[i]:=base[i]-add(v[k]*g(v[k],base[i])/g(v[k],v[k]),
  k=1..i-1) end if;od;
> fc:= proc(n) local a;
  a := array( [ seq (v[i]/f(v[i]),i=1..n) ]);
  return eval( a, 1 ) end proc:e:=fc(n):
> plot(e,t=0..Pi/4);
> A:=Matrix(n,n):
> for i from 1 to n do
```

```
> for j from 1 to n do A[i,j]:=g(e[i],e[j]) od;od;A;
```

E.5 The Compliance Operator

Evaluation of compliance for the system illustrated in Fig.3.2.

```
> restart;with(linalg):
> eq2:=-S+m*diff(diff(v(t),t),t)+nu*diff(v(t),t)+2*k1*
  (x(t)-lambda1(t))*v(t)+2*k2*(l-x(t)-lambda2(t))*v(t):
> eq1:=L-m*diff(diff(x(t),t),t)-nu*diff(x(t),t)-k1*
  (x(t)-lambda1(t))^2+k2*(l-x(t)-lambda2(t))^2:
> L:=4:m:=1:l:=10:nu:=0.1:xi:=nu/m:k2:=0.01:k1:=0.01:
> lambda1(t):=piecewise(t<2, 3, t>=2, 4):
> lambda2(t):=piecewise(t<2, 4, t>=2,5):
> plot([lambda1(t),lambda2(t)],t=0..4,
  colour=[blue,red]):
> s(t):=evalf(dsolve({eq1,D(x)(0)=5,x(0)=0},x(t))):
> s1(t):=convert(rhs(s(t)),list)[2]:
> s2(t):=convert(rhs(s(t)),list)[4]:
> y(t):=piecewise(t<2, s1(t), t>=2, s2(t)):
> plot(y(t),t=0..4):
> epsilon:=1:rho:=Pi/4:f(t):=sin(t):
> g(t):=2*k1*(x(t)-lambda1(t))+2*k2*
  (l-x(t)-lambda2(t)):
> M:=sqrt(12*Pi*c1/c2)*((K+T+Q)/epsilon):
> c1:=evalf(eval(exp(t*xi)*abs(g(t)),t=rho)):
> c2:=minimize(exp(t*xi),t=0..rho):
```

```
> alpha:=maximize(abs(g(t)),t=0..rho):
> beta:=maximize(abs(f(t)),t=0..rho):
> delta:=maximize(abs(diff(f(t),t)),t=0..rho):
> N:=maximize(xi+alpha*abs(t-1),t=0..rho):
> cappa(t):=abs(t)*beta*exp(N*t):
> K:=evalf(maximize(cappa(t),t=0..rho)):
> ti(t):=beta*exp(N*t)*(abs(t)*xi+alpha/N^2*
  abs(t*N-1))+beta*(alpha/N^2+1):
> T:=evalf(maximize(ti(t),t=0..rho)):
> cu(t):=beta*exp(N*t)*(abs(t)*(alpha+xi^2)+(alpha*
  xi)/N^2*abs(t*N-1))+beta*(alpha*xi)/N^2+xi*
  beta+delta:
> Q:=evalf(maximize(cu(t),t=0..rho)):
> evalf(M):n:=15:r:=2*Pi:
> s(t):=rhs(evalf(dsolve({subs(S=f(t),eq2),D(v)(0)=0,
  v(0)=0},v(t)))):plot(s(t),t=0..4):
> for i from 1 to n do wu(i):=piecewise(t>=(i-1)*
  Pi/(4*(n+1)) and t<(i+1)*Pi/(4*(n+1)),-cos(4*(n+1)*
  (t-(i-1)*Pi/(4*(n+1))))+1):od:
> fc:= proc(n) local a;
  a := array( [ seq (wu(i),i=1..n) ]);
  return eval( a, 1 ) end proc:ba:=fc(n):
> nor:=v->sqrt(int(v^2+diff(v,t)^2+
  diff(diff(v,t),t)^2,t=0..Pi/4));
> inn:=(v,w)->int(v*w+diff(v,t)*diff(w,t)+
  diff(diff(v,t),t)*diff(diff(w,t),t),t=0..Pi/4);
> for i from 1 to n do if i=1 then wu[i]:=ba[1]
  else wu[i]:=ba[i]-add(wu[k]*inn(wu[k],
  ba[i])/inn(wu[k],wu[k]),k=1..i-1) end if;od;
```

```
> fc:= proc(n) local a;
  a := array( [ seq (evalf(wu[i]/nor(wu[i])),i=1..n) ]);
  return eval( a, 1 ) end proc:base:=fc(n):
> plot(base,t=0..Pi/4);
> fc:= proc(n) local a;
  a := array( [ seq (subs(v(t)=base[i],x(t)=s(t),eq2-S),
  i=1..n) ]); return eval( a, 1 ) end proc:Stiff:=fc(n):
> with(student):A[n]:=Matrix(n,n):
  for i from 1 to n do for j from 1 to n do
  A[n][i,j]:=inn(Stiff[j],base[i]); od;od;W:=A[n];
> A[n]:det(A[n]):if (det(A[n])=0) then C:=0 else
  C:=evalf[5](inverse(A[n])) fi:W:=A[n]:f:=f(t):
> for i from 1 to n do se(i):=inn(f,base[i]):od:
> fc:= proc(n) local a;
  a := array( [ seq (se(i),i=1..n) ]);
  return eval( a, 1 ) end proc:h:=fc(n);
> z:=multiply(C,h);
> for i from 1 to n do p(i):=base[i]*z[i] od:
> k(t):=evalf(add(p(i),i=1..n)):
> nor(k(t)-sol(t));
```

Acknowledgements

The author wishes to thank the Department of Electric Systems and Automation of the University of Pisa, the Interdepartmental Research Center "E. Piaggio" of the University of Pisa and in particular the Prof. Danilo De Rossi, the Prof. Federico Lorussi and the Prof. Stefano Galatolo.

Bibliography

[1] P Acquistapace. Appunti del corso di Analisi funzionale. http://www.dm.unipi.it/ acquistp/anafun.pdf.

[2] A Ambrosetti and G Prodi. On the inversion of some differentiable mappings with singularities between banach spaces. *Annali di Matematica Pura ed Applicata*, 4(93):231-246, 1972.

[3] A Bicchi Appunti del corso di Robotica. http://www.piaggio.ing.unipi.it/corsi/robotica/teoria-dei-sistemi/unico-text.pdf

[4] E Bizzi, N Accornerro, B Chapple and N Hogan. Posture control and trajectory formation during arm movement. *Journal of Neuroscience*, 4, 27382744, 1984.

[5] H Brezis. Functional Analysis. Theory and Applications. *Liguori Editore*, 1986.

[6] M Caprili, A Cella and G Gheri. Spline interpolation techniques for variational methods. *International Journal for Numerical Methods in Engineering*, Volume 6 Issue 4, Pages 565 - 576, June 2005.

[7] F Carpi D De Rossi. Dielectric elastomer cylindrical actuators: electromechanical modelling and experimental evaluation Matter. *Sci.Eng.* 24 555-62, 2004.

[8] F Carpi and D De Rossi. Dielectric elastomer cylindrical actuators: electromechanical modelling and experimental evaluation. *Materials Science and Engineering*, 24(4):555-562, 2004.

[9] F Carpi and D De Rossi. Dielectric elastomers as electromechanical transducers. *Fundamentals, materials, devices, models and applications of an emerging electroactive polymer technology.*, Elsevier Press, 2008.

[10] H Cartan. Calcul Differentiel. *Hermann*, Paris, 1967.

[11] S Casalini, G Pioggia, M Ferro, C Caudai, and D De Rossi. FACE e la sua mente. *Bioingegneria della Mente, Ingegneria Biomedica.*, Bologna (Italy), 2006.

[12] R Courant and D Hilbert. Methods of Mathematical Physics. Vol. 1. *Wiley-Interscience Editions*, January 1989.

[13] R Courant and D Hilbert. Methods of Mathematical Physics. Vol. 2. *Wiley-Interscience Editions*, January 1989.

[14] D De Rossi, F Di Puccio, F Lorussi, P Orsini, and A Tognetti. Feldman's muscle model: implementation and control of a kinematic chain driven by pseudo-muscular actuators. *Acta of Bioengineering and Biomechanics.*, 4(1):224-225, 2002.

[15] M Dornay, Y Uno, M Kawato, and R Suzuki. Simulation of optimal movements using a 17-muscle model of the monkey's arm. *Proc. SICE 30th Annual Conference.*, 4(1):919-922, Yonezawam Japan, 1991.

[16] A Feldman. Once more on the equilibrium-point hypothesis (λ model) for motor control. *Journal of Motor Behavior*, (18):17–54, 1986.

[17] A Feldman, S Adamovich and M Levin. The relationship between control, kinematic and electromyographic variables in fast single-joint movements in humans. *Experimental Brain Research*, 103, 440-450, 1995.

[18] A Feldman and M Levin. The origin and use of positional frames of reference in motor control. *Behavioral & Brain Sciences*, 18, 723-806, 1995.

[19] A Feldman. Spatial frames of reference for motor control. *Human Kinetics.*, M.L. Latash (Ed.), pp. 289-313, 1998.

[20] A Feldman, A Ostry, M Levin, P Gribble, and A Mitnitski. Recent tests of the equilibrium-point hypothesis. *Motor Control*, num. 2, 26-42, 1998.

[21] M Levin, and A Feldman. The role of stretch reflex threshold regulation in normal and impaired motor control. *Brain Research.*, num.635, 1996.

[22] T Flash and N Hogan. he coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, (7):1688–1703, 1985.

[23] S Brenner and R.L Scott. The Mathematical Theory of Finite Element Methods. *Springer*, 2nd edition, 2005.

[24] P Hopkins. Skeletal muscle physiology. *Academic Unit of Anaesthesia*, Leeds, UK.

[25] Human Physiology http://people.eku.edu/ritchisong/301notes3.htm

[26] F Lorussi, A Tognetti, F Carpi, A Mazzoldi, and D De Rossi. Recruited dielectric elastomer motor units as a pseudomuscular actuator. *Smart Structures and Materials 2003: Electroactive Polymer Actuators and Devices.*, Bar-Cohen Editor, volume 5051, pages 464-467, 2003.

[27] P Morasso. Spatial control of arm movements. *Experimental Brain Research.*, volume 42, pages 223-227, 1981.

[28] Muscle Physiology. University of California. http://muscle.ucsd.edu/

[29] Basic Skeletal Muscle Physiology. http://home.hia.no/ stephens/musfacts.htm

[30] F Mussa-Ivaldi. Modular features of motor control and learning. *Curr.Opin. Neurobiol.*, 9, 713717, 1999.

[31] R Shadmehr and F Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14, 32083224, 1994.

[32] Dr. Otto H. Schmitt Biomimetic Foundation. Official Site. http://www.otto-schmitt.org/

[33] Dr. Otto H. Schmitt Biomimetic Foundation. Memorial Article. http://www.otto-schmitt.org/otto_images/pavekohsbio.pdf

[34] G Pioggia, M Ferro, C Caudai, S Casalini, A Ahluwalia and D De Rossi. Sensory Fusion and Emerging Behaviours in an Anthropomorphic Robot as a Man-machine Interface. *International SenseMaker Workshop on Life Like Perception Systems*, Ireland, 2005.

[35] W Rudin. Functional Analysis. *McGraw Hill*, New York, 1991.

[36] R Schiavi and G Grioli. VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference*, Pasadena, May 2008.

[37] L Sciavicco and B Siciliano. Robotica Industriale. Modellistica e Controllo di Manipolatori. *McGraw Hill*, 1st edition, Milano, 1995.

[38] Y Uno, M Kawato, and R Suzuky. Formation and control of optimal trajectory in human multijoint arm movement: Minimum torque-change model. *Biologocal Cybernetics*, 61:89–101, 1989.

[39] Y Uno, M Kawato, and R Suzuky. Minimum muscle-tension change model which reproduces human arm movement. *Proceedings of the 4th Symposium on Biological and Physiological Engineering*, pp. 299-302, 1989.