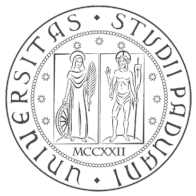


Università degli Studi di Padova

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Magistrale in Ingegneria Meccatronica

Interaction control in a tendon arm mechanism: identification of the environment's parameters

Laureando

Michele Longhi

Relatore

Prof. Roberto Oboe

Co-relatore

Prof. Satoshi Komada

Anno Accademico 2016/2017

Alla mia famiglia che mi ha supportato e sopportato durante questo percorso ricco di momenti difficili seguiti da grandi soddisfazioni.

"Per angusta ad augusta"

Ernesto di Brandeburgo

Abstract

More and more often systems with springs are used to ensure safety and versatility in the surrounding environment in which they are used. That is why studies and research are increased during the years for such mechanisms with non-linear springs, which guarantee greater lightness then reducing the size, therefore cost of the engines.
The first part of this analysis will give an overview of the mechanism called "tendon arm", describing the advantages of implementing this technology. Subsequently, there is described the equipment used and the control technique that allows its movement, called "Force Control".
Next the main point of the thesis will be explained concerning the study of the external environment in which the system is placed. The attention will be focused on the parameters that describe environment, by first making an off-line identification of them, then verifying the improvement/worsening of the system. Finally, the "adaptive control" technique will be implemented, permitting the change of on-line parameters: in this way the system will give a better response in any condition in which it is placed.

Acknowledgement

I would like to thank Professor Roberto Oboe for giving me the opportunity to go to Japan for drawing up this thesis as well as professors Junji Hirai and Satoshi Komada who welcomed me at "MIE University" and, despite the distance, made me feel at home.

Contents

Chapter 1

Introduction

Research into rehabilitation robotics has grown rapidly and the number of therapeutic rehabilitation robots has expanded dramatically during the last two decades.
Rehabilitation robots interact closely with humans, often sharing the same workspace and sometimes physically attaching to humans, as is the case of robotic movement-training devices and prosthetic limbs. Furthermore, the devices are by necessity powerful enough to manipulate the environment or the user's own limbs, which means that they are also often powerful enough to injure the user or another person nearby by colliding with them or moving their limbs inappropriately. *Safety* is clearly of paramount importance.
A common strategy for ensuring safety is to incorporate multiple, redundant safety features. A device can be designed to be mechanically incapable of moving itself or the user's limbs in such a way as to cause injury. Limits can be placed on the range, strength, and speed of actuators so that they can accomplish the desired task but no more. Breakaway attachments can be used to attach to user's limbs. Covers can protect the user from pinch points in the device.
From a systems perspective, when all else fails actively to protect the user, it must be the design itself that makes the robot inherently unable to injure its user. Part of the solution is in reducing the weight, rounding the surface characteristics and making appropriate materials choices.[1].
But designing robots that: can physically interact and move with humans to cooperatively perform motor tasks, physically assist in achieving motor goals, or aid in rehabilitation of movement is a great challenge of robotics. Robots have the potential to improve rehabilitation by initiating treatment earlier than may otherwise be possible, increasing the intensity of training, creating enriched environments that simulate real-world conditions, and allowing patients to practice motor tasks that they may otherwise be unable

to perform alone. However, for robotics to successfully serve such roles in rehabilitation, the interaction between robot and patient should be one that is intuitive and natural. Many previous studies have examined physical interactions between humans and robots in an effort to determine how such robots should be controlled [2] and what roles they should adopt to train, collaborate with, or assist humans in an intuitive and natural fashion. These approaches generally seek to identify desirable features based on intuitive notions of what would work. As a result, rehabilitation robots have largely been implemented in an ad-hoc fashion based on classical control methodologies. For example, many rehabilitation robots are controlled by specifying desired kinematic trajectories that are enforced by moving a subject's joints along a fixed kinematic path. Other control schemes provide assistance on an assist-as-needed basis. Recent advances in human motor learning suggest that humans learn more when errors are larger or when movement variability is greater. This has sparked a new approach in the control of rehabilitation robots; to augment rather than reduce errors. Alternatively, rather than providing assistance or resistance to reduce movement or augment movement errors for an entire kinematic trajectory, rehabilitation robots could be designed to provide assistance or resistance to particular portions of a movement.[3]
To ensure human security, technology is increasingly focus to the use of tendon arm.
In the next sections different type of robots are described in particular *rigid* and *soft* robot.

1.1 Rigid robot

A typical rigid robot is an industrial robot, where it is used for manufacturing. Industrial robots are automated, programmable and capable of movement on two or more axes.
This type of robot can be used for welding, painting, assembly, pick and place for printed circuit boards, packaging and labeling, palletizing, product inspection, and testing; all this is carried out with high endurance, speed, and precision. They can help with material handling and provide interfaces.

1.1.1 Types and features

The most commonly used robot configurations are articulated robots, SCARA robots, delta robots and cartesian coordinate robot. In the context of general robotics, most types of robots would fall into the category of robotic arms. Robots exhibit varying degrees of autonomy:

- some robots are programmed to faithfully carry out specific actions over and over again (repetitive actions) without variation and with a high degree of accuracy. These actions are determined by programmed routines that specify the direction, acceleration, velocity, deceleration, and distance of a series of coordinated motions;

- others robots are much more flexible, as they may need to orientate the objects with which they are working. In addition, some robots have even a sensor to identify the object. For example, for more precise guidance, robots often contain machine vision sub-system acting as their visual sensors, linked to powerful computers or controllers.
  Artificial intelligence is becoming an increasingly important factor in the modern industrial robot.

Figure 1.1: rigid robot: SCARA

1.1.2 Defining parameters

- *Number of axes* - two axes are required to reach any point in a plane; three axes are required to reach any point in a space. To fully control the orientation of the final part of the arm (i.e. the wrist) three more axes (yaw, pitch and roll) are required.
- *Degrees of freedom* - this is usually the same as the number of axes.
- *Working envelope* - the region of space a robot can reach.
- *Kinematics* - the actual arrangement of rigid members and joints in the robot, which determines the robot's possible motions. Classes of robot kinematics include articulated, cartesian, parallel and SCARA.
- *Carrying capacity or payload* - how much weight a robot can lift.
- *Speed* - how fast the robot can position the end of its arm. This may be defined in terms of angular or linear speed of each axis or as a compound speed i.e. the speed of the end of the arm when all axes are moving.
- *Acceleration* - how quickly an axis can accelerate. Since this is a limiting factor, a robot may not be able to reach its specified maximum speed for movements over a short distance or a complex path requiring frequent changes of direction.
- *Accuracy* - how closely a robot can reach a commanded position. When the absolute position of the robot is measured and compared to the commanded position, the error is a measure of accuracy. Accuracy can be improved with external sensing, for example by implementing a vision system or Infra-Red. Accuracy can vary with speed and position within the working envelope and with payload.
- *Repeatability* - the precision with which the robot will return to a programmed position. This is not the same as accuracy. It may be that, when told to go to a certain X-Y-Z position, it gets only within 1 mm of that position. This would be its accuracy, which may be improved by calibration. But if that position is taught into controller memory, and when sent there, it returns within 0.1 mm of the taught position, then the repeatability will be within 0.1 mm.

Accuracy and repeatability are different measures. Repeatability is usually the most important criterion for a robot, and is similar to the concept of

"precision" in measurement. Typically a robot is sent to a taught position a number of times, and the error is measured at every time it returns to the position 4 times in the same point. Repeatability is then quantified using the standard deviation of those samples in all three dimensions. It may happen that a typical robot make a positional error exceeding repeatability, which can be a problem for the system. Moreover, the repeatability is different in distinct parts of the working envelope; it can also changes with speed and payload.
Repeatability in an industrial process is also subject to the accuracy of the end effector, for example a gripper, and even to the design of the "fingers" that match the gripper of the object being grasped. For example, if a robot picks a screw by its hand, the screw could be at a random angle. A subsequent attempt to insert the screw into a hole could easily fail. These and others similar scenarios can be improved by making the entrance to the hole tapered.

- *Motion control* - for some applications, such as simple pick-and-place assembly, the robot need merely return repeatedly to a limited number of pre-taught positions. For more sophisticated applications, such as welding and finishing (spray painting), motion must be continuously controlled to follow a path in space, with controlled orientation and velocity.

- *Power source* - some robots use electric motors, others use hydraulic actuators. The former are faster, the latter are stronger and advantageous in applications such as spray painting, where a spark could set off an explosion; however, low internal air-pressurisation of the arm can prevent ingress of flammable vapours as well as other contaminants.

- *Drive* - some robots connect electric motors to the joints via gears; others connect the motor to the joint directly (direct drive). Using gears results in measurable "backlash" which is the free movement in an axis.

- *Compliance* - this is a measure of the amount in angle or distance through which a robot axis will move when a force is applied to it. Because of compliance when a robot goes to a position carrying its maximum payload, it will be at a position slightly lower than when its carrying no payload. Compliance can also be responsible for overshoot when carrying high payloads. In that case acceleration need to be reduced.

1.1.3 Controlling movement

For a given robot the only parameters necessary to completely locate the end effector (gripper, welding torch, etc.) of the robot are the angles of each of the joints or displacements of the linear axes. However, there are many different ways to define the points. The most common and most convenient way of defining a point is to specify a Cartesian coordinate for it, i.e. the position of the "end effector" in the X, Y and Z directions relative to the robot's origin. In addition, depending on the types of joints a particular robot may have the orientation of the end effector in yaw, pitch and roll and the location of the tool point relative to the robot's faceplate must also be specified. For a jointed arm these coordinates must be converted to joint angles by the robot controller and such conversions are known as Cartesian Transformations which may need to be performed iteratively or recursive for a multiple axis robot. The mathematics of the relationship between joint angles and actual spatial coordinates is called kinematics.[4]

1.2 Robot's work envelope

The operational characteristics of robots can be significantly different from other machines and equipment. Robots are capable of high-energy (fast or powerful) movements through a large volume of space even beyond the base dimensions of the robot. The pattern and initiation of movement of the robot is predictable if the item being "worked" and the environment are held constant. Any change to the object being worked (i.e. physical model change) or the environment can affect the programmed movements.

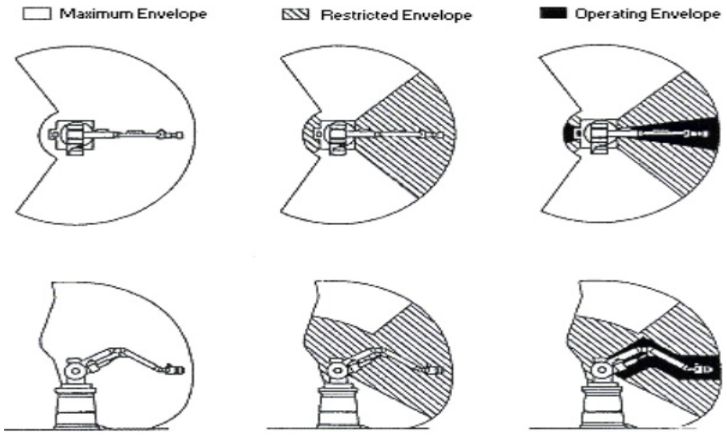


Figure 1.2: A robot's work envelope

Some maintenance and programming personnel may be required to be within the restricted envelope while power is available to actuators. The restricted envelope of the robot can overlap a portion of the restricted envelope of other robots or work zones of other industrial machines and related equipment. Thus, a worker can be hit by one robot while working on another, trapped between them or peripheral equipment.
Additional hazards can also result from the malfunction of, or errors in, interfacing or programming of other process or peripheral equipment. The operating changes with the process being performed or the breakdown of conveyors, clamping mechanisms or process sensors could cause the robot to react in a different manner.
Robotic incidents can be grouped into four categories: a robotic arm or controlled tool causes the accident, places an individual in a risk circumstance, an accessory of the robot's mechanical parts fails, or the power supplies to the robot are uncontrolled.

- *Impact or Collision Accidents.* Unpredictable movements, component malfunctions, or unpredicted program changes related to the robot's arm or peripheral equipment can result in contact accidents.

- *Crushing and Trapping Accidents.* A worker's limb or other body part can be trapped between a robot's arm and other peripheral equipment, or the individual may be physically driven into and crushed by other peripheral equipment.

- *Mechanical Part Accidents.* The breakdown of the robot's drive components, tooling or end-effector, peripheral equipment, or its power source is a mechanical accident. The release of parts, failure of gripper mechanism, or the failure of end-effector power tools (e.g. grinding wheels, buffing wheels, power screwdrivers) are a few types of mechanical failures.

- *Other Accidents.* Other accident can result from working with robots. Equipment that supplies robot power and control represents potential electrical and pressurized fluid hazards. Ruptured hydraulic lines could create dangerous high-pressure cutting streams or whipping hose hazards. Environmental accidents from arc flash, metal splatter, dust, electromagnetic, or radio-frequency interference can also occur. In addition, equipment and power cables on the floor present tripping hazards.

1.2.1 Safeguarding Personnel

For the safeguard of the worker some considerations can be made:

- *Risk Assessment.* At each stage of development of the robot and robot system a risk assessment should be performed. There are different systems and personnel safeguarding requirements at each stage. The appropriate level of safeguarding determined by the risk assessment should be applied. In addition, the risk assessments for each stage of development should be documented for future reference.

- *Safeguarding Devices.* Personnel should be safeguarded from hazards associated with the restricted envelope (space) through the use of one or more safeguarding devices:

  - Mechanical limiting devices;
  - Non-mechanical limiting devices;
  - Presence-sensing safeguarding devices;
  - Fixed barriers (which prevent contact with moving parts);
  - Interlocked barrier guards.[5]

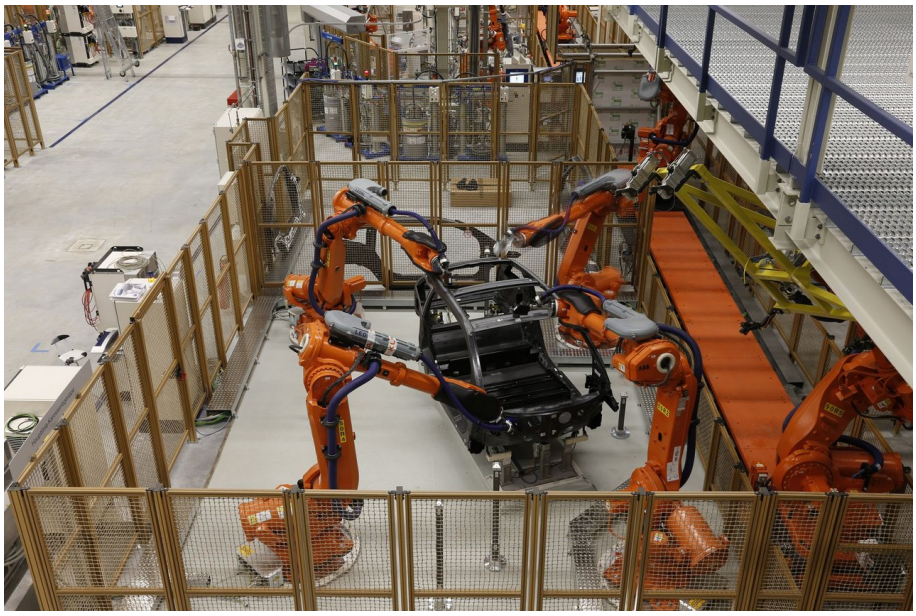


Figure 1.3: Delimitation area for rigid robot

1.3 Soft robot

To avoid the interaction with the workers, technology is becoming more and more towards to the "soft robots".
Soft robotic is the specific sub-field of robotics dealing with constructing robots from highly compliant materials, similar to those found in living organisms. Similarly, soft robotics also draws heavily from the way in which these living organisms move and adapt to their surroundings. In contrast to robots built from rigid materials, soft robots allow for increased flexibility and adaptability for accomplishing tasks, as well as improved safety when working around humans. These characteristics allow for its potential use in the fields of medicine and manufacturing.

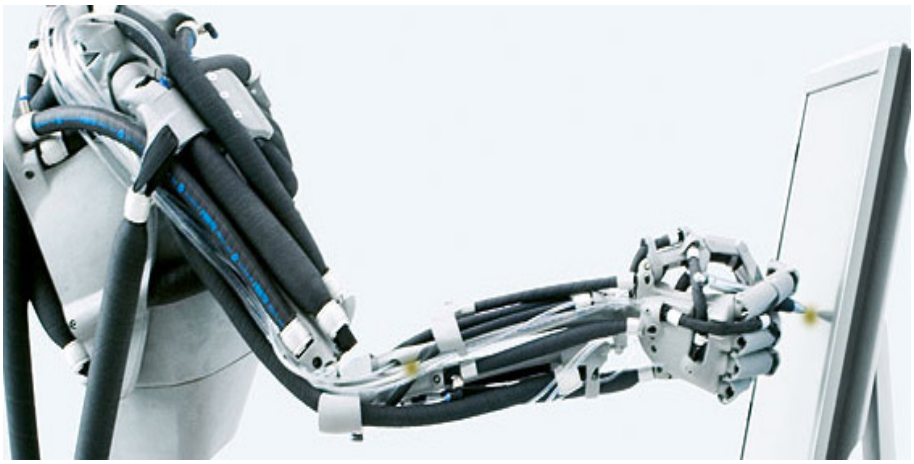


Figure 1.4: Robotic arm

1.3.1 Types and designs

The bulk of the field of soft robotics is based upon the design and construction of robots made completely from compliant materials, with the end result being similar to invertebrates like worms and octopus. The motion of these robots is difficult to model, as continuum mechanics apply to them, and they are sometimes referred to as continuum robots. Soft robotics is the specific sub-field of robotics dealing with constructing robots from highly compliant materials, similar to those found in living organisms. Similarly, soft robotics also draws heavily from the way in which these living organisms move and adapt to their surroundings. In contrast to robots built from rigid materials, soft robots allow for increased flexibility and adaptability for accomplishing tasks, as well as improved safety when working around humans. These

characteristics allow for its potential use in the fields of medicine and manufacturing. However, there exist robots that are also capable of continuum deformations, most notably the snake-arm robot. Also, certain soft robotic mechanics may be used as a piece in a larger, potentially rigid robot. Soft robotic end effectors exist for grabbing and manipulating objects, and they have the advantage of producing a low force that is good for holding delicate objects without breaking them.
Additionally, hybrid soft-rigid robots may be built using an internal rigid framework with soft exteriors for safety. The soft exterior may be multifunctional. It can act as both the actuators for the robot, similar to muscles in vertebrates, and as padding in case of a collision with a person.

1.3.2 Uses and applications

Soft robots can be implemented in the medical profession, specifically for invasive surgery. Soft robots can be made to assist surgeries due to their shape changing properties. Shape change is important as a soft robot could navigate around different structures in the human body by adjusting its form. This could be accomplished through the use of fluid actuation.
Soft robots may also be used for the creation of flexible exosuits, for rehabilitation of patients, assisting the elderly, or simply enhancing the user's strength.
Traditionally, manufacturing robots have been isolated from human due to safety concerns, as a rigid robot colliding with a human could lead to injury due to the fast paced motion of the robot. However, soft robots could work alongside humans safely, as in a collision the compliant nature of the robot would prevent or minimize any potential injury.

1.3.3 Tendon arm

Why is it called "tendon arm"? The answer lies in the anatomy of the human body, in fact, this mechanism simulates the arm of a human.

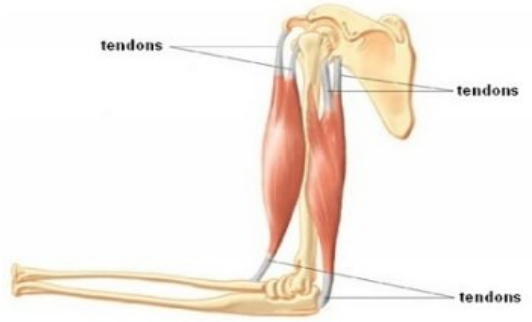


Figure 1.5: Tendon arm

It's important to know that human muscle-skeletal system is antagonistic. The muscle is attached to the skeleton through the tendon, which is responsible for the transfer of force. Note that muscle can only pull, not push. Thus, the antagonism is required. Antagonism means that each joint has at least two muscles, one to flex and the other to stretch the joint. These two muscles work in opposition. Because of this antagonistic actuation, there is a high stiffness during a physical contact. Furthermore, the human muscle-skeletal system has the ability to adjust the stiffness as a function of different parameters. As a result, our joints are non-linear and inherently flexible.
In our case the antagonism is achieved by using two tendons to move a joint, one as flexor and the other as extender. The joint has variable and non-linear stiffness, such as human joints. This is realized by "Variable Stiffness Actuating" (VSA), which consists of several motors, each one equipped with a non-linear springs that reproduce the behavior of a human joint.

Chapter 2

Function of the skeletal muscle

There are three main types of muscles in the human body, called skeletal muscle, smooth muscle and cardiac muscle. When we speak of the movements of the arm, only skeletal muscles are involved.
Skeletal muscles have important functions in our body, especially in the mobility. In fact, they must move the limbs of the body motion, generating and supplying power by creating an active force. An active force is the force (or tension) created by the active element or contractile in response to neural activation. Note that tension may be used in place of strength because they are proportional:

$$Tension = Force \,/\, Cross\ sectional\ area \tag{2.1}$$

In order to generate the movements and/or stability, a chain of actions takes place in the muscular-skeletal system. Skeletal muscles surrounding it, or better wrap, bones and pass through joints. The bones are connected through them. When a muscle is activated by a pulse, it contracts and then creates strength. Consequently, a moment is generated in the joint which will follow a rotational moment, translation, etc.

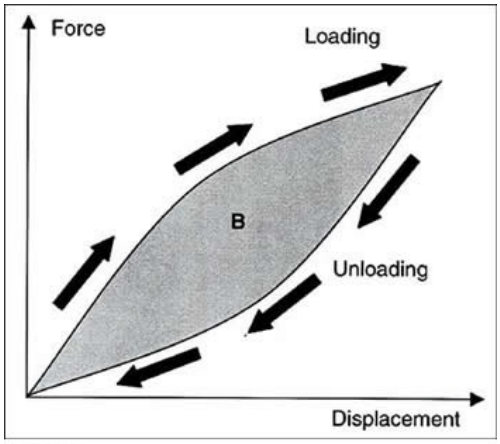
2.1 Properties of skeletal muscle

Generally, the muscles can generate an active force and also, using their *mechanical impedance*, provide the passive force i.e. the resistance to the imposition of motion.
The skeletal muscle properties are divided into:

- **Excitability**. A muscle fibre is capable of responding to stimulate. Due of excitability, the mechanical impedance is also a function of the level of activation. This gives flexibility to our muscular-skeletal system to modify its mechanical characteristics according to the task or to the change of the mechanical properties of the environment.

- **Extensibility and elasticity**. A muscle fibre can be stretched from its rest position, after stretching it returns to its original length. This property refers to *active elasticity* or *active stiffness* of the muscle.

- **Viscoelasticity**. This property is shown by the relation strength-speed which is due to the dependence of muscle strength in connection to the change of muscle's length and vice versa.
  The viscoelasticity derives mainly from the connective tissue, it also has a force-length relation, which can be summarized in the word *passive elasticity*. For example, the tendon's tensile tests show hysteresis during loading/unloading.[6]



Simulated load-deformation curve of viscoelasticity
B = energy lost as a result of friction.

Figure 2.1: Tendon's hysteresis

The characteristics force-length and force-speed are also called as *functional properties* of skeletal muscle. This is due to the fact that the main function of the muscle is to produce force. The functional properties show how well a muscle can create force related to the muscle's length and the speed of contraction.

2.2 Definition of stiffness

In study of the mechanisms, the stiffness is defined as the resistance of an elastic body to the deformation due to an applied force. In other words, the stiffness is the amount of force needed for stretching a body in a unit of length, with N/m the unit of the SI.[7] There are several types of stiffness:

- Based on components:
  - **Muscle stiffness**, **Joint stiffness**, **End-point stiffness**. The joint's stiffness is the amount of torque (Nm) which increases per unit angular deformation of the joint (radiant or degrees).[7] The end point's stiffness is the end of the limb, i.e. hand, foot. The muscle stiffness includes the rigidity of the muscle fibres and connective tissues.
- Based on time:
  - **Static stiffness** and **Dynamic stiffness**. Static stiffness is defined as the difference of the static force to two different lengths. Dynamic one, instead, is the instantaneous stiffness measured during a change in the muscle length. Therefore it represents the derivate of the force respect to the length, which is defined only when the length is changing:

$$K_{din} = \frac{dF}{dL};\ K_{din} \neq 0 \Rightarrow dx \neq 0 \tag{2.2}$$

- Based on the fact that passive elements, such as springs, need an external force to deform:
  - **Passive stiffness**, **Active stiffness**. When an object is deformed by the application of an external force, like a spring, it is called passive stiffness. However, when the stiffness is due to the creation of an active force i.e. muscle contraction, is called active stiffness.

2.3 Stiffness Adjustable Tendon-SAT

The elements that simulate the behaviour of the human tendon are defined: Stiffness Adjustable Tendon (SAT). Since more flex and variable stiffness are required, this research is focused on tendons mechanisms with non-linear springs. They are called *variable stiffness elements* because their stiffness change in relation to their operating point. These particular types of springs have advantages such as light weight and simple structure.
However, the accurate derivation of the characteristic equations of SAT is difficult, due to the hysteresis that modifies the characteristics during elongation and contraction. This causes deterioration of the performance of the controller in tension control.

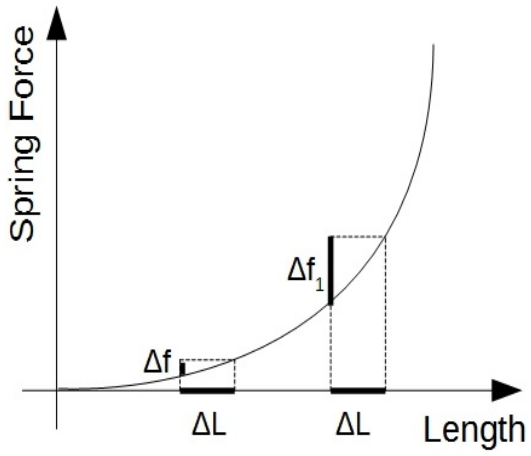


Figure 2.2: SAT characteristic

2.3.1 Characteristic and technology

The technologies of springs based on coil are used in many industries with different structures and characteristics of springs. The SAT are lightweight easy to use. They are made up of a thin sponge rubber to form an elastic body which is inserted in a mesh tube made of polyurethane. The particular shape of the coil spring enables an increase of stiffness during elongation. In addition, the materials of which it is composed, prevent the creation of rust, and the intrusion of external elements in the mesh doesn't affect the performance.

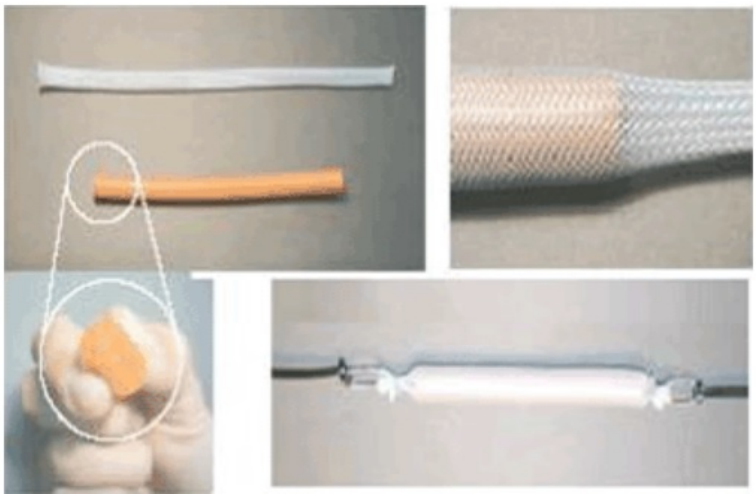


Figure 2.3: Structure of SAT

2.3.2 Equation of Nonlinear Spring SAT

Experiments have revealed approximate equations which allow to understand the functioning of the non-linear springs used in this project. Approximate equation of the i-th SAT, derived using the *least square method*, represents its characteristic ($f_{a,i}(l_i)$) among M SATs is defined by stretching length l_i and constants a_{1i}, a_{2i}, a_{3i} [8]:

$$f_{a,i}(l_i) = a_{1i} \exp^{a_{2i} l_i} + a_{3i} \tag{2.3}$$

The error between $f_{a,i}$ and the real characteristic of i-th SAT $f_{r,i}$ is defined by the approximation error f_i^{dis}:

$$f_i^{dis} = f_{a,i}(l_i) - f_{r,i} \tag{2.4}$$

The vectors of the approximated equation and the approximated error of SAT are defined as follows:

$$f_a(l) = [\ f_{a,1}(l_1) \quad ... \quad f_{a,M}(l_M)\ ]^T \tag{2.5}$$

$$f^{dis} = [\ f_1^{dis} \quad ... \quad f_M^{dis}\ ]^T \tag{2.6}$$

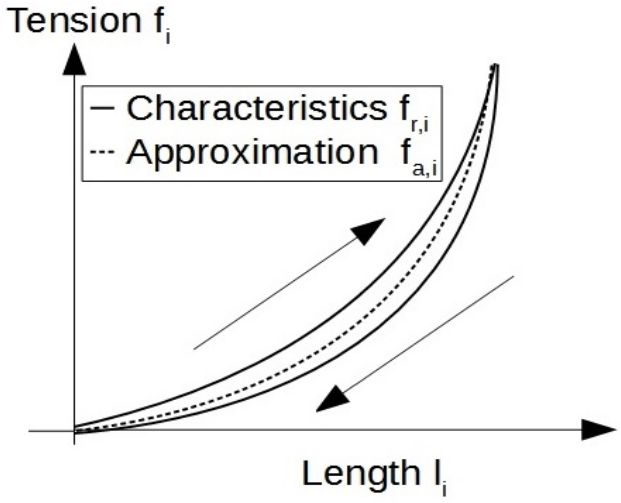


Figure 2.4: Characteristic and approximation of SAT

Chapter 3

SEA and Force Control

3.1 Introduction of the Series Elastic Actuators

Applications with close physical human-robot interaction increase the need for safe robot arms. Introducing elasticity can increase safety when using geared motors because of the decoupling of motor and joint.[9] The degree of elasticity does not necessarily have to be high. Elasticity can be realized by using inherently compliant actuators, such as artificial pneumatic muscles or by using elastic elements in series with standard geared motors. Additional actuators can be used to adapt stiffness or damping, or to combine actuators with complementary features, such as electromotors and pneumatic actuators.
The use of cables in robotic arms can also contribute to safety by allowing to place the motors away from the actuated joints to previous links in the kinematic chain and therefore reducing the link weights.[10] As a drawback, special care has to be taken in the design process regarding the lifespan of the tendons.[11]
Muscles and *tendons* change their stiffness as a function of the motion/task they have to perform. Arm muscles assume a stiff configuration when the arm has to perform an accurate task, while they are compliant when they are performing the "loading" phase of throw. Similarly, if we analyse jumping we see that leg muscles are compliant during the "loading" phase of the jump or during the landing phase where they absorb the shock, while during the "pushing" phase, they are stiff.
There are several reasons for this variation in stiffness, but among the most pressing, we find the exploitation of the elastic energy stored within the muscles and tendons.[12] This enables compliant actuators to achieve performance which is not possible with a conventional stiff robotic system. Clearly

the introduction of compliance may have very significant effects on the performance of an actuation system relative to the classical stiff design.
Sometimes energy saving is considered as a non-priority issue, even though, if autonomous robots are of concern, this feature is one of the most important.

3.2 Standard model of an actuator

A simple model for an actuator with a series elastic element is shown in figure 3.1 below. Shown are the model of the motor mass J_m, the spring with stiffness k_s, the force on the motor T_m, and the output force T_l. The movement of the motor shaft, and the load are θ_m and θ_l respectively.

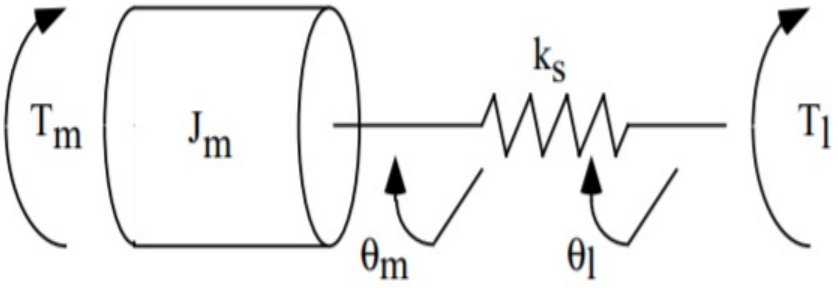


Figure 3.1: Standard model of actuator

Some relations are immediately forthcoming from the diagram, by applying Newton's Laws:

$$T_m + k_s(\theta_l - \theta_m) = J_m\ddot{\theta_m} \tag{3.1}$$

$$-k_s(\theta_l - \theta_m) = T_l \tag{3.2}$$

By taking Laplace transforms, an expression relating T_m and T_l can be found:

$$T_m(s) = (1 + \frac{J_m}{k_s}s^2)T_l(s) + J_m s^2\theta_l(s) \tag{3.3}$$

This equation is important, as it shows what motor torques are needed to give an output torque of T_l, when the output of the actuator is moving.
If the output of the actuator is assumed clamped ($\ddot{\theta_l} = 0$), then the transfer function between output torque, and motor torque is:

$$\frac{T_l}{\theta_l}(s) = \frac{1}{1 + s^2 J_m/k_s} \tag{3.4}$$

Thus the transfer function between the actual output force T_l and the motor force T_m has no zeros, and two poles on the imaginary axis, at a frequency $\omega = \sqrt{K_s/J_m}$ which corresponds to the natural frequency of the motor mass and the spring.
The transfer function between the motion of the output shaft θ_l and the output force T_l can also be written, and is shown below. The ratio T_l/θ_l is also defined as the impedance Z of the system, looking from the output. This transfer function has the same poles as the T_l/T_m but it also has two zeros at the origin. The negative sign comes from the definition of the directions of T_l and θ_l.[13]

$$Z(s) = \frac{T_l}{\theta_l}(s) = \frac{-s^2 J_m}{1 + s^2 J_m / k_s} \tag{3.5}$$

3.3 Antagonistic design

A different type of SEA is used in tendon arm mechanism, in particular the antagonistic design.

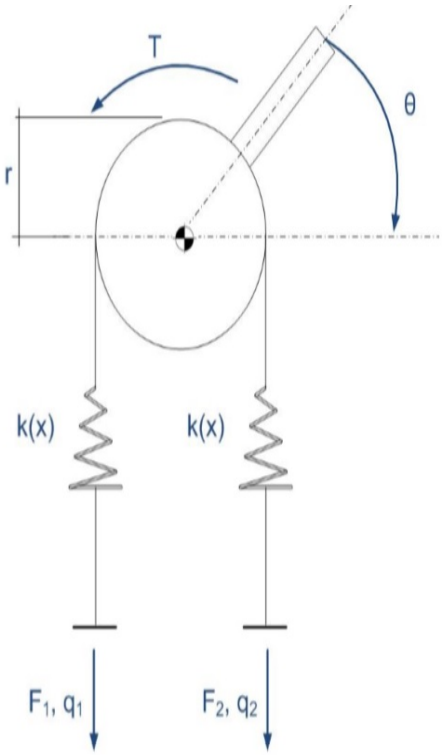


Figure 3.2: Antagonistic set-up

Antagonistically-actuated joints employ two compliant elements to provide power to the joint. This design is biologically-inspired, since mammalian anatomy follows the same concept, i.e. a joint actuated by two muscles arranged in an antagonistic manner. The muscle-tendon cooperation gives the driven link (arm, leg, etc.) a controllable and variable compliance. In addition to biological muscle this type of antagonistic compliance controlled can be achieved using both conventional two motors electric drive designs and other more biologically inspired forms such pneumatic Muscle Actuators (pMA). In the latter case compliance is an inherent characteristic of the actuator, while for an electrical design compliant elements (generally springs) have to be embodied into the system.
An antagonistically-actuated joint uses two driving elements. The stiffness of the joint and the angular displacement of the driven link are set by means of a combination of the actuation inputs q_1, q_2 (see figure 3.2). By co-contraction of these actuators, pre-loading and, thus, tuning of the stiffness is achieved, while the rotation of the joint is obtained by the antagonistic motion of the drives.[14]
It is interesting to derive a simple model of this mechanism. The torque τ applied to the joint, corresponding to a joint angle θ, actuator angular positions θ_1 and θ_2, is given by:

$$\tau = R(k(x)L_1 - k(x)L_2), \tag{3.6}$$

with $k(x)$ the spring stiffness coefficient, $L_1 = r\theta_1 - R\theta$ and $L_2 = r\theta_2 + R\theta$ their elongations, R the radius of the joint pulley, and r the radius of the actuator pulleys.
The effective joint stiffness σ, defined as the infinitesimal variation of the joint torque corresponding to an infinitesimal change of joint angle, while inputs to actuator are held constant (no reliance is thus made on feedback control of actuators in case of impacts), is simply evaluated as:

$$\sigma = \frac{\delta\tau}{\delta\theta} = R^2(2k(x)) - R^2(\frac{\delta k(x)}{\delta L_1}L_1 + \frac{\delta k(x)}{\delta L_2}L_2). \tag{3.7}$$

It can be easily observed that, if stiffness coefficient $k(x)$ is constant, the overall joint stiffness is independent of the actuator inputs. To realize a Variable-Stiffness-Tendon (VST) with this configuration, non-linear elastic elements, depending on the elongation, are therefore in order. In the common case that the two springs are equal and have bounded elongation
$L_{min} \leqslant L_i \leqslant L_{max}$, $i = 1, 2$; we have that the stiffness range (which was shown in the previous section to directly affect performance) evaluates to:[15]

$$\Delta_\sigma = \frac{\sigma_{max} - \sigma_{min}}{\sigma_{max} + \sigma_{min}} = \frac{L_{max} - L_{min}}{\bar{\sigma}}\frac{\delta k}{\delta L} \tag{3.8}$$

3.4 Applications of the Series Elastic Actuators

Active suspension system

One application of Series Elastic Actuators can be applied, for example, for active suspension system. Active suspensions can be used to arbitrarily orient and isolate payloads and can impart restoring forces to a robot's center of mass, providing a dynamic balancing. These active suspension behaviours are not possible when operating a purely position-controlled robot in an unknown environment. For this reason, robots that follow rigid trajectories have trouble walking over rough terrain because they cannot conform to the terrain or apply restoring forces dynamically to the robot. In contrast, Series Elastic Actuators may be used to control the net interaction forces between robot and ground, providing an active suspension system independent from the position of the feet or the roughness of the terrain.

Figure 3.3: Spring Flamingo: walking robot that employ Series Elastic Actuators

Exoskeleton and Haptic Interface

Series Elastic Actuators are ideal for exoskeleton, haptic interface, and other wearable robots. In these applications, the device shouldn't inhibit the natural fluid motion of the user. An exoskeleton should supply assisted forces to the joint during power-intensive activities (such as knee extension when climbing stairs, or carrying heavy loads) without inhibiting motion of the user during low-power activities (such as leg swing and foot placement). A haptic surface should provide force feedback when touching virtual surfaces or performing physical actions, but should not impede the motion when moving

in free space. Ideally, these devices would be perfectly transparent, creating the feeling that the device is not even there. Series Elastic Actuators are a good choice for approximating perfect transparency in these devices because of their low impedance.

Figure 3.4: Example of an Exoskeleton

Robotic Arms

When robotic arms have to operate in uncertain environments, especially when interacting with people, Series Elastic force control can improve performance and safety. By imposing maximum force limits in the joints, operators can work within the workspace of a powerful robot with an added measure of safety. If an operator contacts any point on the robot, a fault will be registered very quickly. Traditional robots would require excessive load sensors throughout the arm to operate safely in an unstructured environment. The inherent shock tolerance of Series Elastic Actuators also allow the robot to crash with minimal or no damage.
Performance can be greatly improved using force control in situations where it's better for the physical constraints of the environment to dictate the motion of the robot as opposed to a preprogrammed trajectory. [16]

Figure 3.5: Robotic arm

3.5 Force Control

3.5.1 Introduction of force control

A fundamental requirement for the success of a manipulation task is the capability to handle the physical contact between a robot and the environment. Pure motion control turns out to be inadequate because the unavoidable modelling errors and uncertainties may cause a rise of the contact force, ultimately leading to an unstable behaviour during the interaction, especially in the presence of rigid environments. Force feedback and force control becomes mandatory to achieve a robust and versatile behaviour of a robotic system in poorly structured environments as well as safe and dependable operation in the presence of humans.
Research on robot force control has flourished in the past three decades. Such a wide interest is motivated by the general desire of providing robotic systems with enhanced sensory capabilities. Robot using force, touch, distance, and visual feedback are expected to autonomously operate in unstructured environments other than the typical industrial shop floor.
Since the early work on telemanipulation, the use of force feedback was conceived to assist the human operator in the remote handling of objects with a slave manipulator. More recently, cooperative robot systems have been developed where two or more manipulators (for example the fingers of a dexterous robot hand) are to be controlled so as to limit the exchanged forces and avoid squeezing of a commonly held object. Force control plays a fundamental role also in the achievement of robust and versatile behaviour of robotic systems in open-ended environments, providing intelligent response in unforeseen situations and enhancing human-robot interaction.

3.5.2 From Motion Control to Interaction Control

Control of the physical interaction between a robot manipulator and the environment is crucial for the successful execution of a number of practical tasks where the robot end-effector has to manipulate an object or perform some operation on a surface. Typical examples in industrial settings include polishing, deburring, machining or assembly. A complete classification of possible robot tasks, considering also non industrial applications, is practically infeasible in view of the large variety of cases that may occur, nor would such a classification be really useful to find a general strategy to control the interaction with the environment.
During contact, the environment may set constraints on the geometric paths that can be followed by the end-effector, denoted as *kinematic constraints*.

This situation, corresponding to the contact with a stiff surface, is generally referred to as *constrained motion*. In other cases, the contact task is characterized by a dynamic interaction between the robot and the environment that can be inertial (as in pushing a block), dissipative (as in sliding on a surface with friction) or elastic (as in pushing against an elastically compliant wall). In all these cases, the use of a pure motion control strategy for controlling interaction is prone to failure.
Successful execution of an interaction task with the environment by using motion control could be obtained only if the task was accurately planned. This would in turn require an accurate model of both the robot manipulator (kinematics and dynamics) and the environment (geometry and mechanical features). A manipulator model may be known with sufficient precision, but a detailed description of the environment is difficult to obtain, even if in this thesis it's explained a method to estimate the environment.
To understand the importance of task planning accuracy, it is sufficient to observe that in order to perform a mechanical part mating with a positioning of the parts should be guaranteed with an accuracy of an order of magnitude greater than part mechanical tolerance. Once the absolute position of one part is exactly known, the manipulator should guide the motion of the other with the same accuracy.
In practice, the planning errors may give rise to a contact force and moment, causing a deviation of the end-effector from the desired trajectory. On the other hand, the control system reacts to reduce such deviations. This ultimately leads to a build-up of the contact force until saturation of the joint actuators is reached or breakage of the parts in contact occurs.
The higher the environment stiffness and position control accuracy are, the more easily a situation like the one just described can occur. This drawback can be overcome if a *compliant behaviour* is ensured during the interaction. This compliant behaviour can be achieved either in a passive or in an active interaction control.

Passive Interaction Control

In passive interaction control the trajectory of the robot end-effector is modified by the interaction force due to the inherent compliance of the robot. The compliance may be due to the structural compliance of the links, joints, and end-effector, or to the compliance of position's servo. Soft robot arms with elastic joints or links are purposely designed for intrinsically safe interaction with humans. In industrial applications, a mechanical device passive compliance, known as the remote center of compliance device, is widely adopted. This is a compliant end-effector mounted on a rigid robot, designed and op-

timized for peg-into-hole assembly operations.
The passive approach to interaction control is very simple and cheap, because it doesn't require force/torque sensors; also, the preprogrammed trajectory of the end-effector mustn't be changed at execution time; moreover, the response of a passive compliance mechanism is much faster than active repositioning by a computer control algorithm. However, the use of passive compliance in industrial applications lacks flexibility, since for every robotic task a special-purpose compliant end-effector has to be designed and mounted. Also, it can only deal with small position and orientation deviations of the programmed trajectory. Finally, since no forces are measured, it can't guarantee that high contact forces will never occur.

Active Interaction Control

In active interaction control, the compliance of the robotic system is mainly ensured by a purposely designed control system. This approach usually requires the measurement of the contact force and moment, which are feedback to the controller and used to modify or even generate on-line the desired trajectory of the robot end-effector.
Active interaction control may overcome the aforementioned disadvantages of passive interaction control, but it's usually slower, more expensive, and more sophisticated. To obtain a reasonable task execution speed and disturbance rejection capability, active interaction control has to be used in combination with some degree of passive compliance: feedback, by definition, always comes after a motion and force error has occurred, hence some passive compliance is needed in order to keep the reaction forces below an acceptable threshold.

Force Measurement

For a general force-controlled task, six force components are required to provide complete contact force information: three translational force components and three torques. Often, a force/torque sensor is mounted at the robot wrist, but other possibilities exist, for example, force sensors can be placed on the fingertips of robotic hands; also, external forces and moments can be estimated via shaft torque measurements of joint torque sensors. However, the majority of the applications of force control (including industrial applications) is concerned with wrist force/torque sensors. In this case, the weight and inertia of the tool mounted between the sensor and environment (i.e. the robot end-effector) is assumed to be negligible or suitably compensated from the force/torque measurements. The force signals may be obtained using

strain measurements (e.g. optically), resulting in a compliant sensor. The latter approach has an advantage if additional passive compliance is desired.

From Indirect Force Control to Hybrid Force/Motion Control

Active interaction control strategies can be grouped into two categories: those performing indirect force control and those performing direct force control. The main difference between the two categories is that the former achieve force control via motion control, without explicit closure of a force feedback loop; the latter instead offer the possibility of controlling the contact force and moment to a desired value, thanks to the closure of a force feedback loop.
To the first category belongs *impedance control*, where the deviation of the end-effector motion from the desired motion due to the interaction with the environment is related to the contact force through a mechanical impedance with adjustable parameters. A robot manipulator under impedance control is described by an equivalent mass-spring-damper system with adjustable parameters. This relationship is an impedance if the robot control reacts to the motion deviation by generating forces, while it corresponds to an admittance if the robot control reacts to interaction forces by imposing a deviation from the desired motion. Special cases of impedance and admittance control are *stiffness control* and *compliance control*, respectively, where only the static relationship between the end-effector position and orientation deviation from the desired motion and the contact force and moment is considered. Notice that, in the robot control literature, the terms impedance control and admittance control are often used to refer to the same control scheme; the same happens for stiffness and compliance control. Moreover, if only the relationship between the contact force and angular velocity is of interest, the corresponding control scheme is referred to as *damping control*
Indirect force control schemes don't require, in principle, measurements of contact forces and moments; the resulting impedance or admittance is typically non-linear and coupled. However, if a force/torque sensor is available, then force measurements can be used in the control scheme to achieve a linear and coupled behaviour.
Differently from indirect force control, direct force control requires an explicit model of the interaction task. In fact, the user has to specify the desired motion and the desired contact force and moment in a consistent way with respect to the constraints imposed by the environment. A widely adopted strategy belonging to this category is *hybrid force/motion control*, which aims at controlling the motion along the unconstrained task directions and force (and moment) along the constrained task directions. The starting

point is the observation that, for many robotic tasks, it is possible to introduce an orthogonal reference frame, known as the compliance frame (or task frame) which allows one to specify the task in terms of natural and artificial constrains acting along and about the three orthogonal axes of this frame. Based on this decomposition, hybrid force/motion control allows simultaneous control of both the contact force and the end-effector motion in the mutually independent subspaces. Simple selection matrices acting on both the desired and feedback quantities serve this purpose for planar contact surfaces, whereas suitable projection matrices must be used for general contact tasks, which can also be derived from the explicit constraint equations. Several implementation of hybrid motion control schemes are available, e.g., based on inverse dynamics control in the operational space, passivity-based control, or outer force control closed loops around inner motion loops, typically available in industrial robots.
If an accurate model of the environment is not available, the force control action and the motion control action can be superimposed, resulting in a *parallel force/position control* scheme. In this approach, the force controller is designed so as to dominate the motion controller; hence, a position error would be tolerated along the constrained task directions in order to ensure force regulation.[17]

3.5.3 Inverse Kinematic

Some kinematic issues arise with motion control of robot manipulators, especially in force-based motion control approach based on the relationship between end-effector forces and generalized joint forces.
An important kinematic issue associated with motion control of robot mechanism is the *inverse kinematic problem* or more generally the *task transformation problem*. This problem is raised by the discrepancy between the world where tasks are specified and the world where motions are controlled.
Tasks are specified with respect to the robot's end-effector or manipulated object, while motions are typically controlled through the action of servo-controllers that effect the positions and velocities of the robot's joints. Finding the set of joint trajectories, inputs to the joint servo-controllers, that would produce the specified task is the central issue in the task transformation problem.
Obviously, the need for solutions of the inverse kinematic problem is not limited to the motion control problem. The inverse kinematic is needed in workspace analysis, design, simulations and planning of robot motions. By its computational complexity, however, the inverse kinematic problem becomes more critical in real-time control implementations. This is, for instance, the

case of tasks where the robot is called to accommodate motion that cannot be pre-planned or to make corrections generated by external sensory devices. The wide use of position-based motion control is partly a natural result of the state-of-the-art in manipulator mechanical technology. Current manipulator technology relies almost exclusively on the concept of joint position control, whereas a prerequisite to force-based motion control implementation is the manipulator's ability to achieve precise control of joint torques. This ability, however, is considerably restricted by non-linearities and friction inherent in the actuator-transmission systems generally used in most industrial robots. These limitations have, in addition, a major impact on the dynamic performance that can be achieved. Despite the theoretical advances in manipulator control, PID (Proportional-Integral-Derivative) controllers are still largely dominant in industrial robot system. With PID controllers each joint is independently controlled. The dynamic interaction between joints is ignored, and the disturbance rejection of the dynamic forces relies on the use of large gains and high servo rates.
Robot joint torque control ability is essential not only for achieving higher dynamic performance, but also for the implementation of many force-based part mating operations. Active force control which has emerged as one of the basic means to extend robot capabilities also requires joint torque control capability.
In recent years, there has been an important effort to close the gap between the technologies in robot mechanisms and robot control. Recent trends and ongoing development, e.g. Direct-Drive arms, the ARTISAN manipulator and micro-manipulator system. With the emergence of new capabilities for effective joint torque control, force-based motion control will become a natural control modality for robot manipulators.

Kinematic Control

The computation complexity of the inverse kinematic problem has led to solutions based on the inverse of the linearised kinematic model. This model expresses the relationship between the vector δq associated with the variations of the positions and orientations of the end-effector:

$$\delta x = J(q)\delta q \tag{3.9}$$

where $J(q)$ is the Jacobian matrix. For a n-degree of freedom manipulator with an end-effector operating in an m-dimensional space, $J(q)$ is an n x m matrix.

Force-Based Motion Control

In the resolved motion-rate control approach, the inverse kinematic problem is replaced by a computationally less difficult problem which involves solving a system of linear equations. With the force-based motion control approach, the whole issue of task transformation is eliminated. End-effector motions are directly controlled by forces and moments acting along or about the directions where the task is described. These forces and moments are created by the application of a set of generalized joint forces. The basic relationship between end-effector forces and joint forces is given by

$$\Gamma = J^T(q)F. \tag{3.10}$$

For robot mechanisms where joint torque control capability is available, force-based motion control is clearly the most appropriate approach to be taken. In this approach, control forces can be easily designed to achieve the coordination of joint motions for many complex tasks (goal position, trajectory and surface tracking, collision and joint limit avoidance), whereas only elementary displacement of the end-effector are allowed in the linearised inverse kinematic model.[18]

3.5.4 Models for Force Control

Traditionally, models for robot dynamic have been derived from principles of rigid body dynamics. As a result, the force control problem has not been formulated as a dynamic control problem. When rigid body models are employed for the robot along with ideal actuator models, the actuator inputs are related to the positions through a second order differential equation. However, the output forces are algebraically related to the actuator forces (inputs) and therefore the formulation is devoid of dynamics. In other words, there is a lack of causality in the relationship between the output forces and the inputs. At best the actuators can *compensate* for forces in a static manner, but they cannot dynamically control the forces. In fact, theories for compliance (or stiffness) control, operational space control, hybrid control and their extension to system with closed chains have the same limitations. In the past these theories proved to be adequate because the focus was on performing complex tasks in a quasi-static framework as opposed to dynamic control, but dynamic control is not possible with this approach.
While rigid body models are justified when robots are position controlled, interactions with dynamic environments cannot be controlled with control laws derived from such models. The basic idea is recognize by Hogan and Colgate.[19] They suggest that an impedance model for the contact be used

for control system design. Seering and Eppinger [20] study problems that arise due to non co-location of sensors and actuators in force control schemes. In particular, they also consider dynamic models for the actuator, sensor and environment.

Conventional Approaches to Force Control

Consider the force control problem of a robot system with n degrees of freedom. If I is the n x n inertia matrix, J is the Jacobian and τ is the vector of actuator forces/torques, then the dynamic equations of motion can be written as:

$$I(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + J^T F = \tau \tag{3.11}$$

where C is a non-linear function of positions and velocities. This equation is of the form:

$$f(\theta, \dot{\theta}, \ddot{\theta}) + J^T F = \tau \tag{3.12}$$

If we decompose the input τ into τ_m and τ_f such that

$$f(\theta, \dot{\theta}, \ddot{\theta}) = \tau_m \tag{3.13}$$

$$J^T F = \tau_f \tag{3.14}$$

it is possible to design a position control scheme with τ_m as the input and a force control scheme with τ_f as the input. At the best, the actuators can *compensate* for external forces in a *static* manner.
Alternatively, in the state-space formulation:

$$\begin{cases} \dot{x} = A(x)x + B(x)u \\ y = C(x)x + D(x)u \end{cases} \tag{3.15}$$

where x, y and u are the state, output and input respectively. (For example, $x = [\theta^T, \dot{\theta}^t]^T$, $y = F$ and $u = \tau$).[21]

3.5.5 Force control in a SEA

Series Elastic Actuators provide many benefits in force control in robots place in free environments. These benefits include high strength, low impedance, low friction and good bandwidth in force control. These characteristics are desired in many applications such as robotic legs; exoskeletons to improve humans performance; robotic arms and adaptive suspension. SEAs implement a new mechanical architecture that goes against the principle of "Stiffer is better".

In traditional industrial operations, robots must perform repetitive work at high speed and accuracy. In this setting, where the environment is controlled and tasks are repetitive, the position control with predefined trajectories is an optimal solution. However, in special "environments" where environment is almost unknown, a strong control that compensates this problem is better. This is, for example, the case of robotic legs walking on ruined land; robotic arms that have to interact with people; exoskeletons to improve performance and other applications.
An actuator with an ideal force control should be a perfect source, giving in output the reference force regardless of load movements. In real world, all force's actuators have limitations that cause a deviation from the perfect source of force. These limitations include the impedance, friction and bandwidth. *Impedance* is a function of frequency of the movement of load, usually increases with the frequency of movement of the load. *Friction* can be solved with a breakaway force, which limits are found in the small force that the actuator can give in output. *Bandwidth* of an actuator is the frequency above which the forces can be controlled with accuracy. The bandwidth is affected by saturation of the power elements, and mechanical stiffness of the control system gain in addition to other things.
In a perfect force's source, impedance is zero, friction is zero and bandwidth is infinite. The muscle has very low impedance, moderate friction and bandwidth, this currently makes it the best actuator which provides force.
Series Elastic Actuators have low impedance and friction, thus, it can achieve high quality force control. Therefore, they are suitable for robots in unstructured environments. In Series Elastic Actuators, stiff load cells (which are delicate, expensive and induce chatter) are replaced with a significantly compliant elastic element (which is robust, inexpensive and stable). Figure 3.6 shows architecture of SEA. Note that Series Elastic Actuators are topologically similar to any motion actuator with a load sensor and closed loop control system.

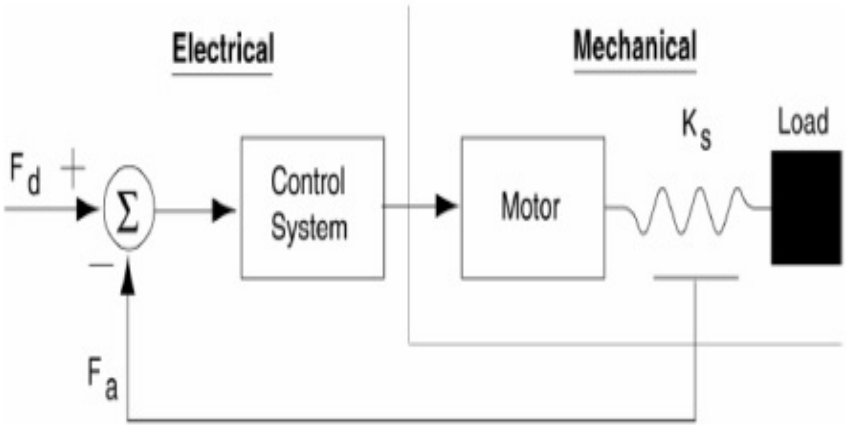


Figure 3.6: Schematic diagram of a Series Elastic Actuators

As we can see in figure 3.6, a spring is placed between the motor and the load. A control system is implemented to reduce the GAP between the desired force and the measured force signal.
Similar to the load cell method, Series Elastic Actuators use active force sensing and closed loop control to diminish the effects of friction and inertia. By measuring the compression of the compliant element, the force on the load can be calculated using Hooke's Law, which states that:

```
elongation and applied force are directly proportional to each
other, and the constant of proportionality is named
elastic constant K of the spring with unit of measure N/m.
```

Listing 3.1: Hooke's Law

$$F = K * \Delta l \tag{3.16}$$

A feedback controller calculates the error between the actual force and the desired force, applying appropriate current to the motor to correct any force errors.
In contrast to the load cell method, Series Elasticity introduces significant compliance between the actuator's output and the load, allowing for greatly increased control gains. Consider, as above, the case of a compliant spring between a linear actuator and rigid load. A moderate linear movement will generate a very small force reading. Thus, closed loop control gains can be very high while still insuring the absence of chatter and presence of stability. Increased control gains greatly reduce impedance (increase back-driveability) and reduce the effects of friction giving the actuators clean force output. Because high impedance and high friction components are tolerable, weight and cost can be reduced by using smaller, less precise actuator components and

replacing expensive load cells with simple springs and position transducers (encoder, potentiometers).
Compared to an actuator with a stiff load cell, Series Elastic Actuators have the following benefits:

1. The actuators exhibit lower output impedance and back-driveability. The dynamic effects of the motor inertia and gear train friction are nearly invisible at the output. In traditional system, the actuator dynamic often dominate the mechanism dynamics, making it difficult to accomplish tasks that require high force fidelity.

2. Shock tolerance is greatly improved by the spring placed in series between motor and load.

3. The force transmission fidelity of the gear reduction is no longer critical, allowing inexpensive gear reduction to be used. Gears typically transmit position with much higher fidelity than force. The series elasticity serves as a transducer between gear reduction output position and load force, greatly increasing the fidelity of force control.

4. The motor's required force fidelity is drastically reduced, allowing inexpensive motors to be used. It is the motor shaft's position, not its output torque that is responsible for the generation of load force. As a result, motors with large torque ripple can be used.

5. Force control stability is improved, even in intermittent contact with hard surface. Chatter is eliminated since a relatively large spring deflection is required to exert a small force.

6. Energy can be stored and released in the elastic element, potentially improving efficiency in harmonic applications. Animals commonly utilize the elasticity of tendons to store energy in one part of a locomotive cycle and release it in another, with the muscle doing much less work overall than would otherwise be required. Series Elastic Actuators may allow the same effect to happen in robots, exoskeleton, or other applications, thereby extending their range.

7. The actuator has lower passive impedance at high frequencies. Traditional actuators have impedance that resembles a large inertia (the motor's rotor inertia multiplied by the square of the gear ratio) at high frequencies. A Series Elastic Actuator looks like a spring at high frequencies, which is much more forgiving of collisions and other unexpected interactions.

Chapter 4

Tendon Arm

4.1 Introduction

In a human living area, robots need to be safe and a versatile when performing various tasks. In order to meet these requirements there is a mechanism named "tendon arm" that use non-linear springs. This arm assures the safety by using the springs into the mechanism. Moreover, a mechanical stiffness caused by the springs in the mechanism can be variable by using the non-linear springs. These springs, named Stiffness Adjustment Tendon-SAT, make this system lighter, cheaper and with a simple structure. It's an important mechanism because in recent years, a working area of the robot has been expanding from a restricted area such as factories to a human living area such as hospitals and offices. Thereby, two performances have been required for the robots which aim to coexist with human. One is the *safety* for contact motions with human and environment, the other is the *versatility* for various tasks. General robots have heavy and rigid arms because actuators are directly mounted at the joints. Therefore, these robots cannot have an adequate safety.[22]

The arm with non-linear springs has the safety and versatility because of the light-weight arms and the variable mechanical stiffness.
However, derivation of accurate equation for SAT characteristic is difficult because of its hysteresis which has different characteristics for stretching and shrinking. This yields deterioration of control performance of a tension controller using the equations of non-linear spring characteristic.

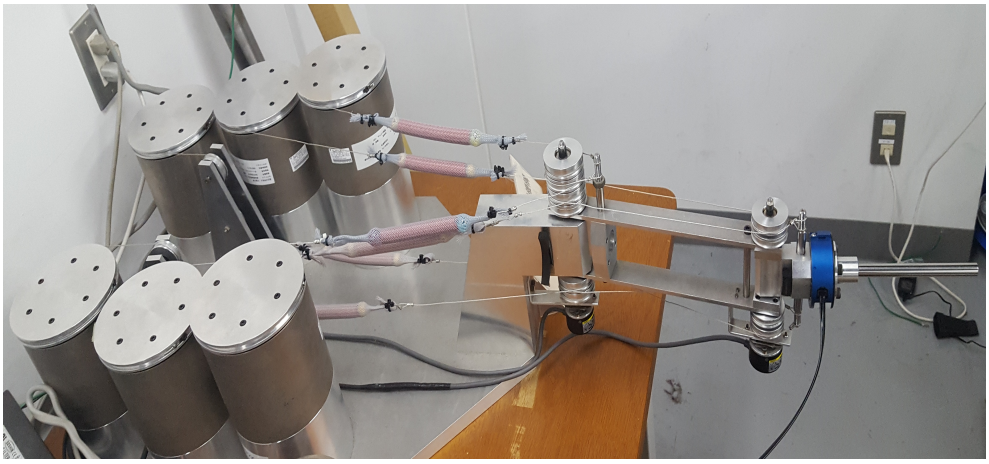


Figure 4.1: Tendon arm

4.2 Tendon Arm with Non-linear Springs

For this research, the tendon arm has been considered with **one degree of freedom** instead of two degrees, so the second joint (where the force sensor has been placed) was kept fixed. Figure 4.2 shows a diagram of the one-joint tendon arm. θ_m, τ_m, and r_m are the rotation angle, the torque and the radius of the motors, respectively. θ_j, τ_j, and r_j are the rotation angle, the torque and the radius of the joint, respectively. f is the force of the springs. In this figure two motors and the arm are connected with the wire trough two springs. Two spring forces are produced by motors, and the torque is generated by the difference of the two spring forces. Therefore, absorbing an impact to the arm is attained by the springs. Furthermore, the mechanical stiffness of the springs becomes variable by using non-linear springs.

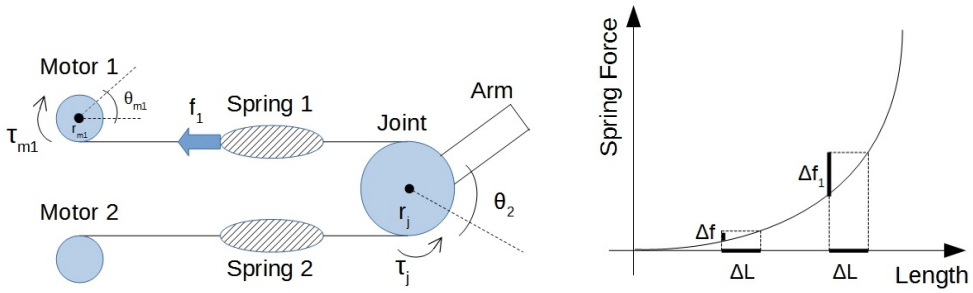


Figure 4.2: One joint tendon arm and Spring characteristic

4.3 Modelling of Tendon Arm

This section formulates the model of the tendon arm with N joints and M tendons.

4.3.1 Kinematics

Length of non-linear springs l is calculated by

$$l = r_m\theta_m + J_j\theta_j \tag{4.1}$$

From virtual work principle, the joint torque τ_j is expressed by using the Jacobian matrix J_j and the spring force f as

$$\tau_j = -J_j^T f \tag{4.2}$$

4.3.2 Jacobian Matrix of Tendon Arm

The tendon arm of the figure 4.1, can be schematized as in figure 4.3. Therefore, to find the corresponding equations:

$$\begin{cases} x = l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ y = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \end{cases} \tag{4.3}$$

$$\Downarrow \frac{d}{dt}$$

$$\begin{cases} v_x = \dfrac{dx}{dt} = -l_1\dot{\theta}_1\sin\theta_1 - l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ v_y = \dfrac{dy}{dt} = l_1\dot{\theta}_1\cos\theta_1 + l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \end{cases} \tag{4.4}$$

In vector form: $\begin{bmatrix} v_x \\ v_y \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$;

where J =Jacobian matrix= $\begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$

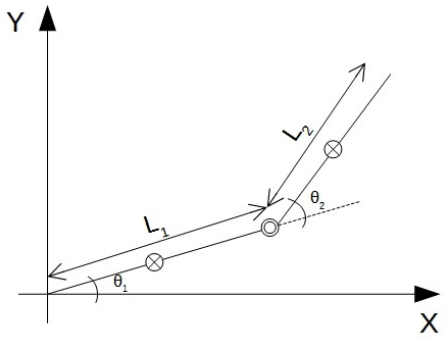


Figure 4.3: 2-link manipulator

4.3.3 Motion Equations of Tendon Mechanism

For calculations, the reader is invited to go to **Appendix A**.
In a compact form the results are:

$$\tau_j(\Theta_j) = M(\Theta_j)\ddot{\Theta}_j + h(\Theta_j, \dot{\Theta}_j)\dot{\Theta}_j \tag{4.5}$$

where

$$M(\Theta_j) = \begin{bmatrix} m_1 l_{g1}^2 + m_2 l_1^2 + m_2 l_{g2}^2 + I_1 + I_2 + 2m_2 l_1 l_{g2} \cos\Theta_2 & m_2 l_{g2}^2 + I_2 + m_2 l_1 l_{g2} \cos\Theta_2 \\ m_2 l_{g2}^2 + I_2 + m_2 l_1 l_{g2} \cos\Theta_2 & m_2 l_{g2}^2 + I_2 \end{bmatrix}$$

$$h(\Theta_j, \dot{\Theta_j}) = \begin{bmatrix} -m_2 l_1 l_{g2} (2\dot{\Theta}_1 + \dot{\Theta}_2)\dot{\Theta}_2 \sin\Theta_2 \\ m_2 l_1 l_{g2} \dot{\Theta}_1^2 \sin\Theta_2 \end{bmatrix}$$

$$g(\theta) = \begin{bmatrix} (m_1 g l_{g1} + m_2 g l_1)\cos\theta_1 + m_2 g l_{g2} \cos(\Theta_1 + \Theta_2) \\ m_2 g l_{g2} \cos(\Theta_1 + \Theta_2) \end{bmatrix}$$

4.3.4 Non-linear Spring SAT

As said in Chapter 3, SAT spring has some advantages as the structure is: lightweight, simple, cheap, rustproof, and etc.[23] SAT has hysteresis characteristic that causes different characteristics for stretching and shrinking.

The approximated equation of i-th SAT among M SATs $f_{a,i}(l_i)$ is derived by a least square method and is defined by stretching length l_i and constants a_{1i}, a_{2i} and a_{3i}.

$$f_{a,i}(l_i) = a_{1i}e^{a_{2i}l_i} + a_{3i} \tag{4.6}$$

The error between $f_{a,i}(l_i)$ and the real characteristics of i-th SAT $f_{r,i}$ is defined by the approximation error f_i^{dis} as:

$$f_i^{dis} = f_{a,i}(l_i) - f_{r,i} \tag{4.7}$$

The vectors of the approximated equation and the approximation error of SAT are defined:

$$f_{a,i}(l) = \begin{bmatrix} f_{a,1}(l_1) & .. & f_{a,M}(l_M) \end{bmatrix}$$
$$f^{dis} = \begin{bmatrix} f_1^{dis} & .. & f_M^{dis} \end{bmatrix}.$$

4.3.5 Joint Stiffness

By deriving the equation (4.6) with respect to the length l_i the stiffness z_i is defined as

$$z_i = \frac{\delta f_{a,i}}{\delta l_i} = a_{2i}f_{a,i} - a_{2i}a_{3i} = c_i f_{a,i} + d_i \tag{4.8}$$

where $c_i = a_{2i}$ and $d_i = -a_{2i}a_{3i}$. Let,

$$K = diag\{z_i, \quad i = 1, .., M\}. \tag{4.9}$$

Here, joint stiffness matrix K_j is expressed by

$$K_j = J_j^T K J_j \tag{4.10}$$

K_j is a NxN positive definite symmetric matrix, and a vector s_j is constructed by components of upper triangular portion of K_j. Thus, s_j is defined by the following equation.

$$s_j = col\{k_{uv}, \quad u, v = 1, .., N, \quad v \geq u\}. \tag{4.11}$$

Here k_{uv} shows the $u, v - th$ element of k_j. Since s_j is linear in association with z_i, the joint stiffness vector s_j is expressed by

$$s_j = Hz. \tag{4.12}$$

From (4.12) can be rewritten by

$$s_j = H(cf + d). \tag{4.13}$$

Here, $c = diag\{c_1, .., c_M\}$, $d = [d_1, .., d_M]^T$. From (4.2) and (4.12), relation among joint torque τ_j, joint stiffness s_j, and tendon tension f is expressed by

$$\begin{bmatrix} \tau_j \\ s_j - Hd \end{bmatrix} = \begin{bmatrix} J_j^T \\ Hc \end{bmatrix} f = J_a^T f. \tag{4.14}$$

where J_a^T is a $N(N+3)/2xM$ matrix called *extended Jacobian matrix*.

4.4 Control System

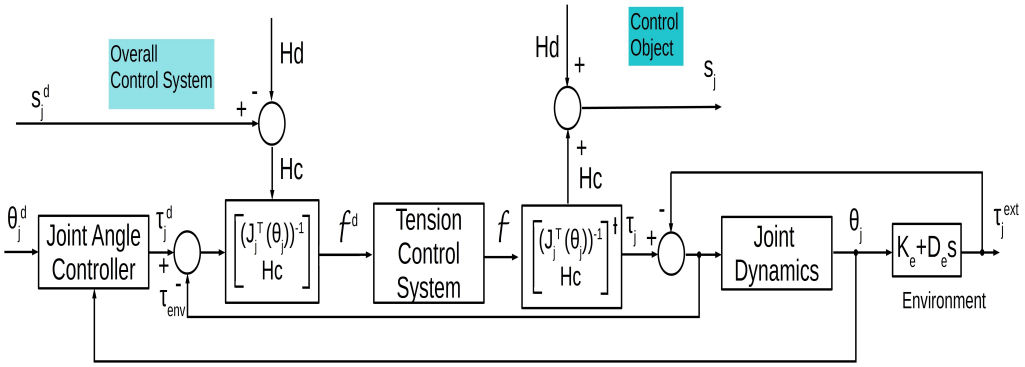


Figure 4.4: Control system of tendon mechanism

Figure 4.4 shows the block diagram of the control system for tendon mechanisms using non-linear springs. The joint torque command derived by the joint controller τ_j^d and the stiffness command $s_j^d - Hd$ are converted to the tension reference f^d by the pseudo-inverse of the extended Jacobian matrix $(J_a^T)^{-1}$ shown in the equation (4.14). Since response of joint stiffness control depends on response of tension control, the tension controller requires high tracking performance.

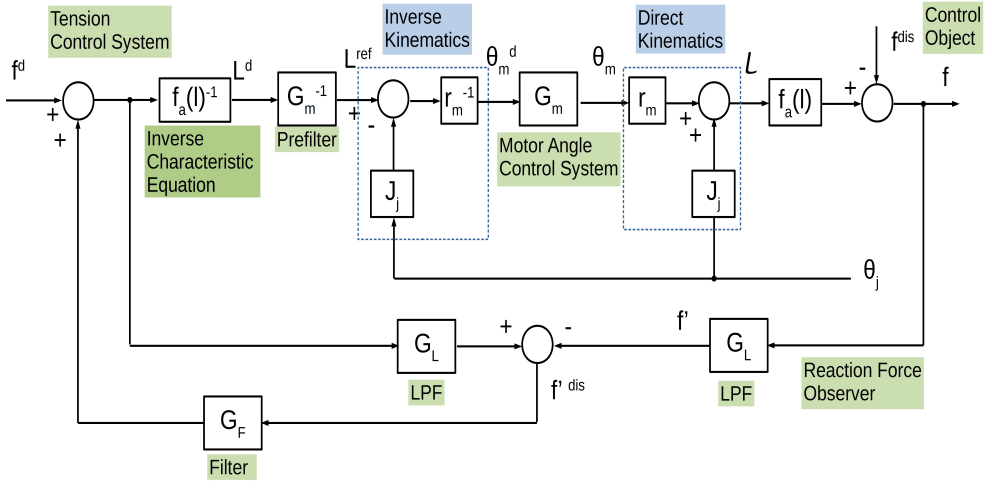


Figure 4.5: Tension control method

As a result of the hysteresis characteristic of the non-linear spring SAT used in this research, it's difficult to obtain high tracking performance. Figure 4.5 shows the tension control system where the inverse function of the approximated spring characteristic $f_{a,i}(l_i)^{-1}$ shown in (4.6) is employed as $f_i(l_i^{-1})$. The approximation error f_i^{dis} shown in (4.7) influences on tension control. Therefore, the approximation error f^{dis}, i.e. error due to approximation of spring characteristics, is estimated by the *disturbance observer* and is compensated by its feedback.

4.4.1 Equivalent Tension Control

Since the transfer function between two non-linear elements from l^d to l becomes unitary by the pre-filter G_m^{-1}, the disturbance observer is able to estimate the approximation error f^{dis}. Here, the tension for feedback f is estimated by the reaction force observer from the motor dynamics to realize the tension sensorless system [24].

$$f = r_m^{-1}\{\tau_m - M_m\ddot{\theta_m} - b_m(\theta_m, \dot{\theta_m})\}. \tag{4.15}$$

Since the diagonal transfer function matrix of low pass filters G_L is used in the reaction force observer, the same low pass filter G_L is inserted to the other input of the disturbance observer to coincide phase each other. The same low pass filter to G_L is inserted to the other input of the disturbance observer to coincide phase each other. By feeding back the estimated disturbance f'^{dis}, the error of the approximated spring characteristics f^{dis} is compensated. The diagonal transfer function matrix G_F in the feedback loop is to ensure properness of G_m^{-1}.

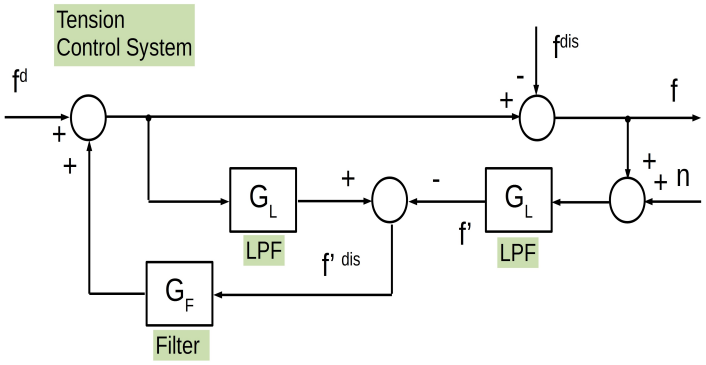


Figure 4.6: Equivalent tension control method

When the transfer function from l^d to l is unitary the system is simplified as shown in figure 4.6, where n represents the noise from the reaction force estimation shown in (4.15), which arises from sensor noise of motor encoders. The tension response of the system is given by

$$f = f^d - (1 - G_{LF})f^{dis} - G_{LF}n \tag{4.16}$$

where $G_{LF} = G_L G_F$. From (4.16), the filter G_{LF} should have low pass characteristics from disturbance suppression and sensitivity to noise point of view, whose cut-off frequency is determined by considering a trade-off between the disturbance and the noise characteristics.

4.4.2 Motor Angle Control

Motor angle control system is shown in figure 4.7. Since the tension f becomes the disturbance for motors, as shown in the motor dynamics represented in (4.15), its compensation f_{comp} is performed. In the method using the tension command f^d, the equation of spring characteristics $f_a(l)$ and the estimated tension f' as f_{comp}, the performance relies on the tension control performance, the accuracy of the equation and its estimation accuracy, respectively.

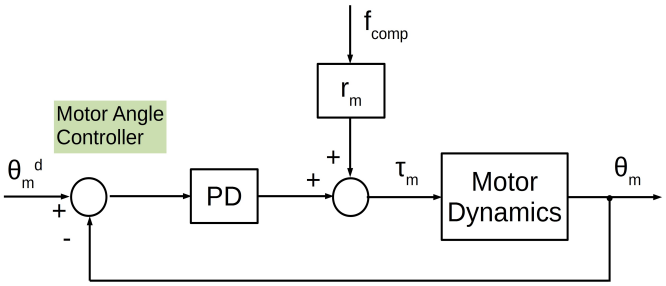


Figure 4.7: Motor angle control system

4.4.3 Joint Angle Controller

The last control part is about the joint angle. In figure 4.8 $\tau(\theta_j^d)$ represents a *feed-forward compensator* derived by the joint dynamics (4.18).

$$\tau(\theta_j^d) = M_j(\theta_j^d)\ddot{\theta}_j^d + \{\frac{1}{2}M_j(\theta_j^d) + S_j(\theta_j^d, \dot{\theta}_j^d)\}\dot{\theta}_j^d + g_j(\theta_j^d) + b_j(\theta_j^d, \dot{\theta}_j^d). \tag{4.17}$$

$$\tau_j^{dis} = M_j(\theta_j)\ddot{\theta}_j + \{\frac{1}{2}M_j(\theta_j) + S_j(\theta_j, \dot{\theta}_j)\}\dot{\theta}_j + g_j(\theta_j) + b_j(\theta_j, \dot{\theta}_j) + J_j^T f \tag{4.18}$$

Here, M_j is an arm inertia matrix, S_j is a skew-symmetric matrix, g_j is a gravity vector, τ_j^{dis} is a joint disturbance torque vector from the environment. The gains of the PD controller are decided so that the pole of the system is $-25rad/s$. The value of the joint stiffness is calculated by substituting the estimated tension f' into (4.13).

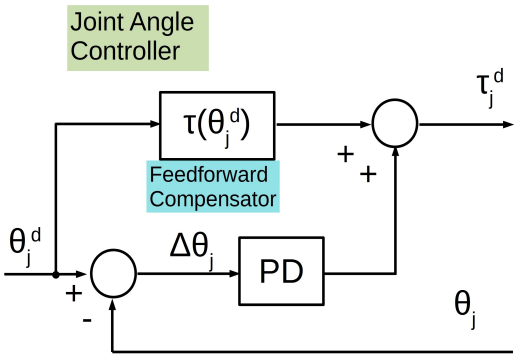


Figure 4.8: Joint angle controller

All tendon arm can be summarize by the *value define map* in figure 4.8. The mechanism is divided into circles, each one representing a part of it. The arrows indicate how to obtain a new information thanks to the known constants.

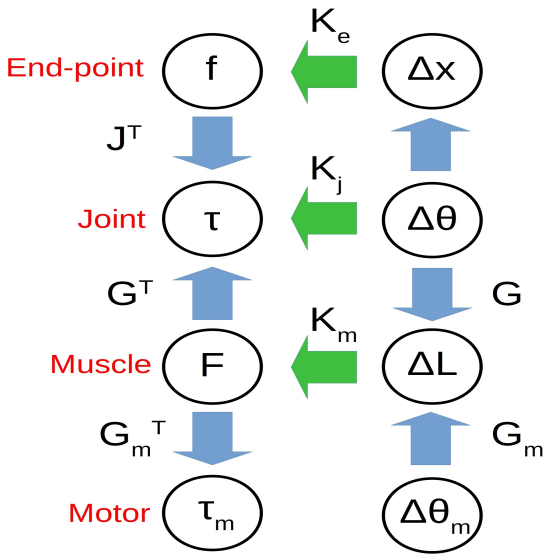


Figure 4.9: Value define map

Chapter 5

Off-line identification

5.1 Introduction

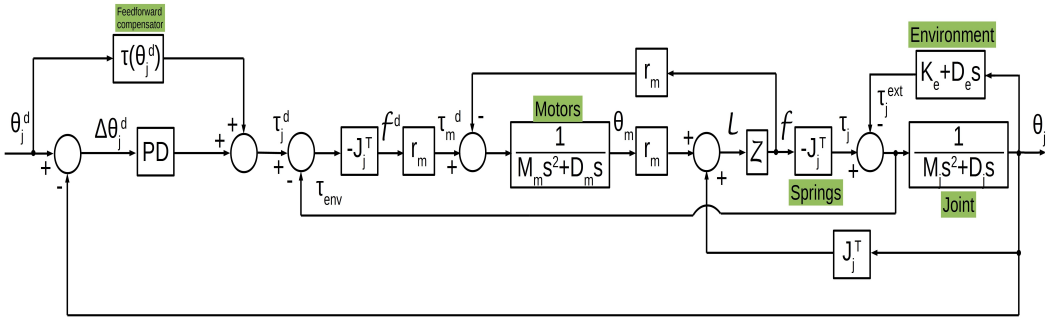


Figure 5.1: Position controller for tendon arm in off-line identification

In figure 5.1 some parts of the control system have been omitted in order to make the explanation as clear as possible.
Nothing has changed about the behaviour of the control explained in Chapter 4: the reference position of the joint θ_j^d is converted to joint torque reference τ_j^d. After, τ_j^d is converted to the spring force command f^d by the pseudo-inverse Jacobian matrix $(-J_j^T)^{-1}$. Moreover, the spring force command f^d is transformed to the motor torque command τ_m^d by the radius of the motors r_m. To obtain the estimated force f, the estimated length l of the springs is multiplied to the mechanical stiffness Z, represented by a linear constant spring stiffness z_i.[25]

$$Z = diag\{z_1, .., z_n\}$$

For the off-line identification part, the external torque τ_j^{ext} is found by the force sensor placed at the end of the arm, while the joint's position θ_j is

found by the encoder place to the joint.

5.2 Identification method

Identification is a method through which dynamic models from a measured data can be estimated. There are some "ingredients" to solve the problem of identification:

1. **Data**

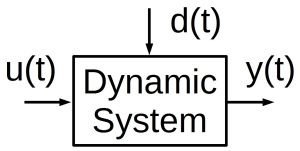


Figure 5.2: Elements of the dynamic system

- $u(t)$ is the measured input, in the case under examination the joint's position θ_j;
- $y(t)$ is the measured output, so the external torque τ_j^{ext};
- $d(t)$ is the disturbance from external causes impossible to measure;
- *dynamic system* is the transfer function representing the environment.

2. **Class of models**
   Some models can be considered in order to obtain the best estimation from the stored data. The most common are:

   - ARX $(n_a,\, n_b)$: $\sum_{i=0}^{n_a} a_i y(t-i) = \sum_{i=0}^{n_b} b_i u(t-i) + e(t)$, with $a_0 = 1$
     $\Rightarrow A(z^{-1})y(t) = B(z^{-1})u(t) + e(t)$ with $e(t) =$ white noise.
   - ARMAX $(n_a,\, n_b,\, n_c)$: $\sum_{i=0}^{n_a} a_i y(t-i) = \sum_{i=0}^{n_b} b_i u(t-i) + \sum_{i=0}^{n_c} c_i e(t-i)$, with $c_0 \neq 0$ and $a_0 = 1$
   - Output Error (OE): $y(t) = F(z)u(t) + e(t)$ where
     $F(z) = \dfrac{B(z^{-1})}{A(z^{-1})}$, $A(z^{-1}) = \sum_{i=0}^{n_a} a_i z^{-i}$, $B(z^{-1}) = \sum_{i=0}^{n_b} b_i z^{-i}$

   In this thesis Output Error is used, where $F(z)$ is the transfer function of the environment and $e(t)$ is the white noise.

3. **Estimation criteria**
   In order to have a good estimation, the estimator must have the following properties:

   - *UNBIASEDNESS*: $E[\hat{\Theta}_N] = \Theta$;
   - *CONSISTENCE*: $\lim_{N\to+\infty} \hat{\Theta}_N = \Theta$;
   - *"small" VARIANCE/QUADRATIC ERROR*: $Var\{\hat{\Theta}_N\} \to 0$ and $E\{\|\hat{\Theta}_N - \Theta\|^2\} \to 0$.

5.3 Estimation by using MATLAB toolbox

As said before, to obtain the estimation of the environment's parameters it is necessary to know the joint position θ_j found by the encoder place to the joint, and the external torque τ_j^{ext} found by the force sensor placed at the end of the arm.

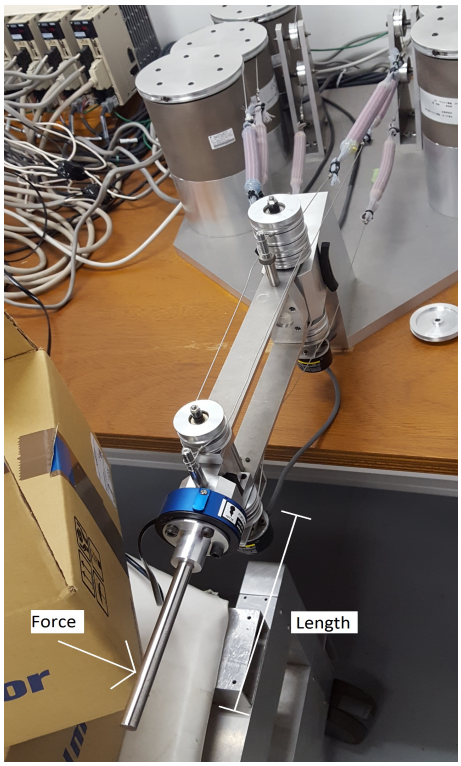


Figure 5.3: Force measurement by using force sensor

An obstacle is placed to measure the external force. Multiplying the measured force for the length, the external torque τ_j^{ext} is found.

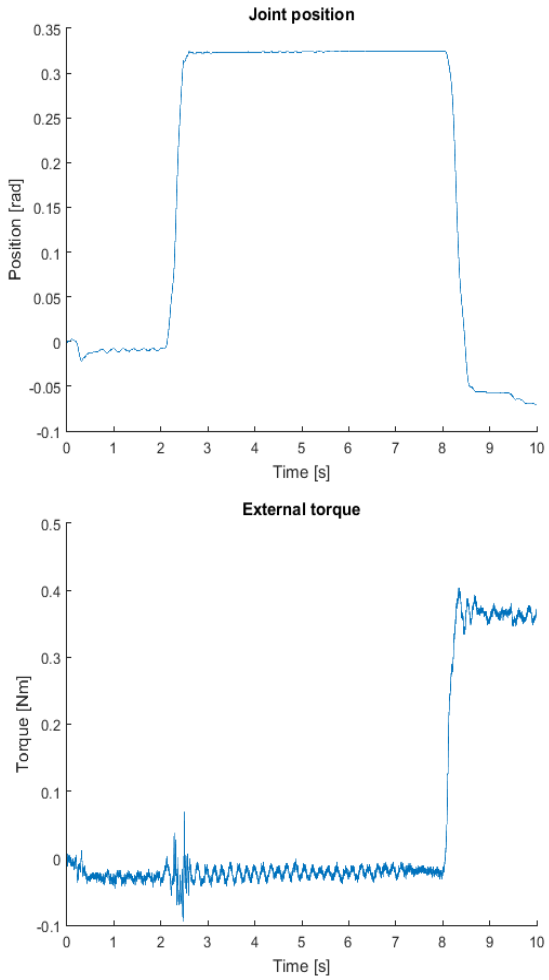


Figure 5.4: Joint position θ_j and External torque τ_j^{ext}

Once these two data (joint position θ_j and external torque τ_j^{ext}) have been found and saved them in a text file, "identification MATLAB toolbox" is used to derive the transfer function representing the external environment. To estimate this transfer function some steps have to be followed:

- write in the command window of MATLAB the instruction *ident*;
- insert the input θ_j and the output τ_j^{ext} data saved in the text file and put them in a vector;
- choose the transfer function to estimate.

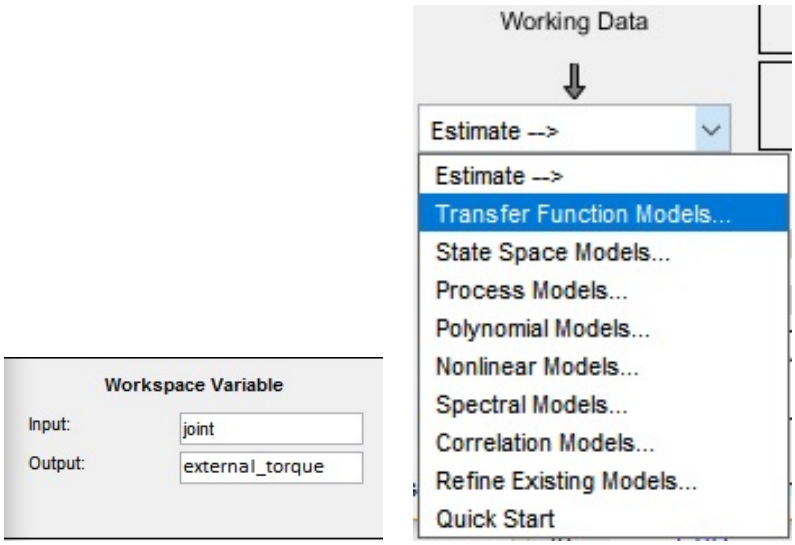


Figure 5.5: Steps to follow for MATLAB identification toolbox

By watching "System Identification Toolbox User's Guide", the method used to estimate the transfer function is *Recursive Prediction Error* (RPE). This method is the basis of the *Recursive Least Square* algorithm used later on, and is defined as:

$$J(t) = arg\,min\{\frac{1}{N}\sum_{i=1}^{N}(y(t) - \hat{y}(t \mid t-1))^2\} \tag{5.1}$$

where $J(t)$ is the function cost that RPE tries to minimize, $y(t)$ is the output and $\hat{y}(t \mid t-1)$ is the prediction. The terms in parentheses are named *Prediction Error*=$\varepsilon(t)$

5.4 Estimation by using Recursive Least Square algorithm

To verify the correctness of the method just explained, the *Recursive Least Square* (RLS) algorithm is performed. The operations can be summarized by the equations:

$$\hat{\alpha}(t) = \hat{\alpha}(t-1) + k(t)[y(t) - \gamma^T(t)\hat{\alpha}(t-1)] \tag{5.2}$$

$$\begin{aligned} k(t) &= P(t)\gamma(t) = \\ &= P(t-1)\gamma(t)[\beta I + \gamma(t)^T P(t-1)\gamma(t)]^{-1} \end{aligned} \tag{5.3}$$

$$P(t) = \frac{1}{\beta}[I - k(t-1)\gamma^T(t)]P(t-1) \tag{5.4}$$

which provide the estimate $\hat{\alpha}$ that minimizes the function cost:

$$J(\alpha, t) = \frac{1}{2}\sum_{i=1}^{t} \beta^{t-i}[y(t) - \gamma^T(i)\alpha(t)]^2 \tag{5.5}$$

where $0 < \beta \leq 1$ is called *forgetting factor*. In the previous equations, $t = kT$ with T is the *sampling time*.
The *exponential forgetting* β is useful for:

- *tracking* time-varying parameters;
- *recovering* a correct estimate by starting from an imprecise initial estimation.

It can be determined as:

$$\beta = e^{-T/T_f} \tag{5.6}$$

where T_f is the *time constant* of the exponential forgetting, i.e. the time after which old data is weighted less than 37% with respect to new data.

T_f	T	β
0.1 s	1 ms	≈ 0.99
1 s	1 ms	≈ 0.999
10 s	1 ms	≈ 0.9999

Table 5.1: Example of forgetting factor

By using saved data of joint's position θ_j and external torque τ_j^{ext}, RLS algorithm is used in off-line mode. The table 5.1 summarizes the environment's parameters found with MATLAB toolbox and with Recursive Least Square algorithm.

	MATLAB toolbox	RLS algorithm
$K_e[Nm/rad]$	3.7819	3.821
$D_e[kgm^2/s]$	3.8482	3.658

Table 5.2: Identified environment's parameters with MATLAB toolbox and with RLS algorithm

Parameters are very similar with both methods, which confirm that MATLAB identification toolbox works properly.
Further, the next steps are to perform Simulation and Experiment, in order to obtain the response of the system.

5.5 Simulations

After off-line identification, simulation was made in order to verify the response of the tendon arm. For this purpose "SIMULINK" is used.
To do simulation the block diagrams explain in Chapter 4 are used in order to obtain the simulated joint position θ_j and the simulated stiffness s.

Results of the simulation

Before put in the real system the identified parameters, a simulation is carried out to see how the system's respond in a *simulated tendon arm.*
Comparisons are made with old parameters founded in the previous simulation and the estimated parameters founded by using MATLAB toolbox.

	Old	New
$K_e[Nm/rad]$	1000	3.7819
$D_e[kgm^2/s]$	10	3.8482

Table 5.3: Old and new identified environment parameters

Figure 5.6 shows the stiffness of the 6 springs, "ref" is the stiffness reference and "sim" indicates the simulated stiffness. Figures show that reference and simulation stiffness are almost the same but in particular the peak at twelfth second is disappeared, allowing a better response of the system, with less undesirable changes.

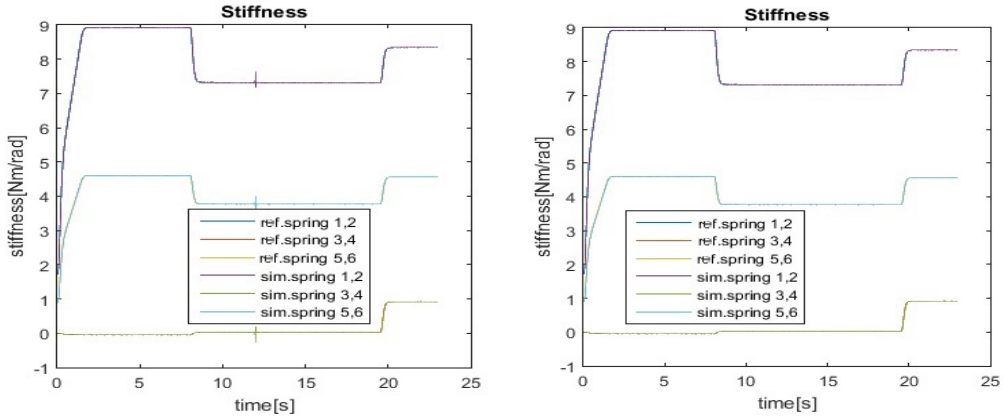


Figure 5.6: Stiffness with old and new parameters, respectively

In Figure 5.7 we can see the joints reference position "ref" and joints simulated position "sim". In the second figure, simulated position is closer to the reference position, therefore tendon arm has a better tracking of the reference position that makes the mechanism more precise. These results show that by simulation the system improves.

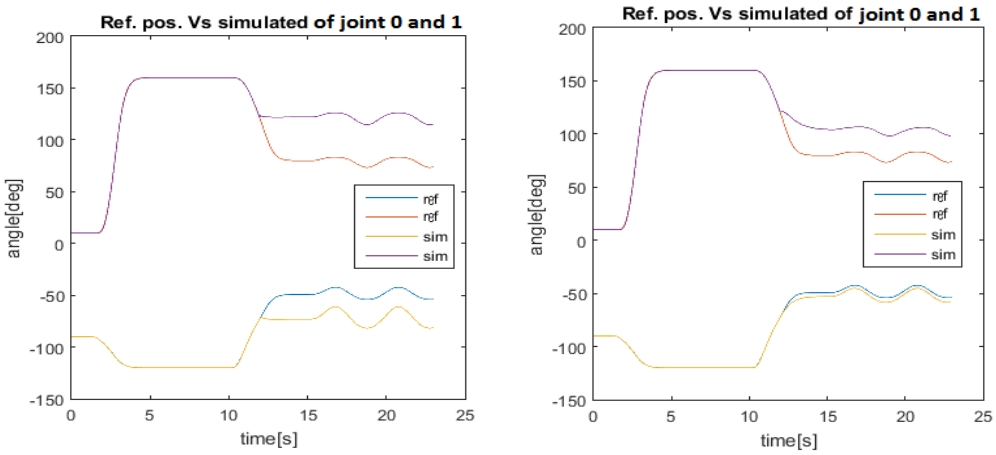


Figure 5.7: Joint position with old and new parameters, respectively

5.6 Experiment

In this part the aim is to verify experimentally the correctness of the estimated parameters.
The estimated torque of the arm, is obtained by the equation:

$$\tau_j(\Theta_j) = M(\Theta_j)\ddot{\Theta}_j + h(\Theta_j, \dot{\Theta}_j)\dot{\Theta}_j \tag{5.7}$$

While the external torque τ_j^{ext} is calculated by multiplying parameters estimated previously with position of the arm found by the encoder attached to it.
To see in the frequency domain the trend of tendon arm, the transfer function from θ_j^d to θ_j of the figure 5.8, is found. The parameters utilized in the transfer function are summaries in table 5.4.

Parameter	Meaning	Value
M_{M1}	Mass motor 1	0.0078
M_{M2}	Mass motor 2	0.0080
D_{M1}	Viscosity motor 1	0.003
D_{M2}	Viscosity motor 2	0.0052
r_M	Radius motor 1,2	0.0675
M_j	Mass joint	1.3
D_j	Viscosity joint	0.0053
Z	Stiffness of spring	20
D_e	Environment parameter	3.8482
K_e	Environment parameter	3.7819
K_p	Controller parameter	70.1064
K_d	Controller parameter	4.9021
$J_j[0]$	Jacobian matrix	0.0245
$J_j[1]$	Jacobian matrix	0

Table 5.4: Parameters of control system

Bode diagrams in figures show: without the feedback of τ_{env}, the magnitude at low frequency has a difference from 0 dB, meaning that joint position θ_j is different from reference joint position θ_j^d; this error is not present in the Bode diagram with the feedback of τ_{env}. Also, the peak of the magnitude in the second figure is disappeared, which was causing the oscillations shown in figures 5.10 and 5.11.

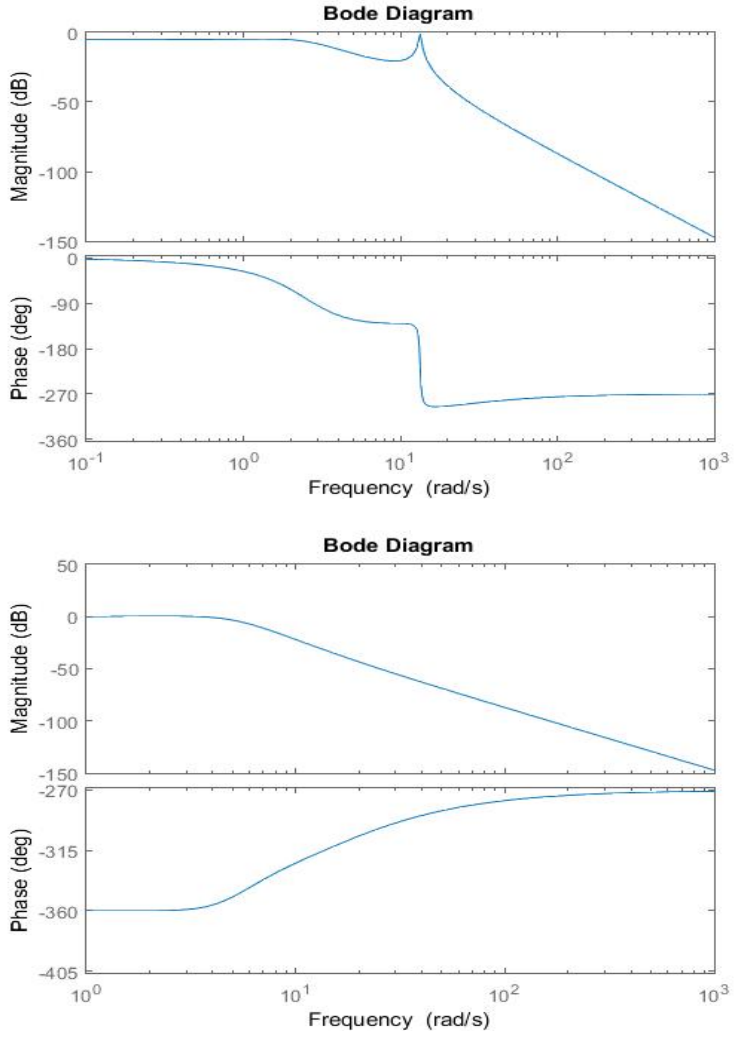
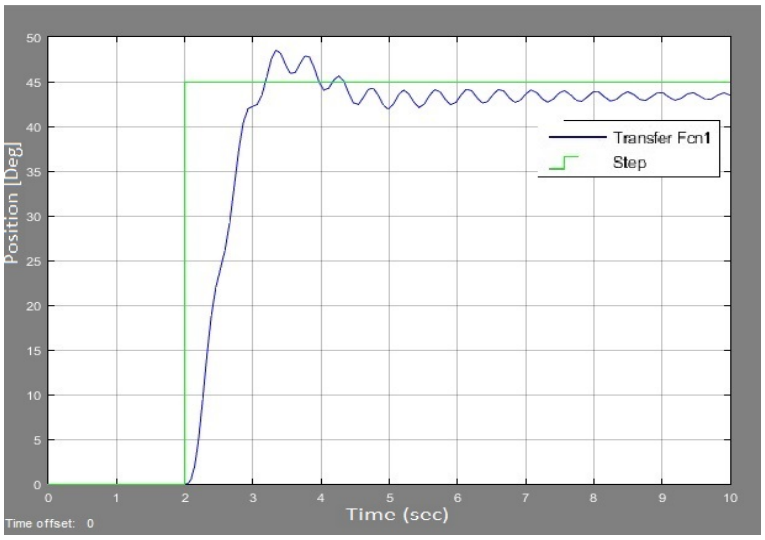


Figure 5.8: Bode diagram from θ_j^d to θ_j without and with feedback of τ_{env}, respectively

In addition a simulation is made in order to see the response of the two transfer functions. Figures 5.10 show the simulation of the transfer function, without and with the feedback τ_{env}, respectively. As we can see, without the feedback from reference position to simulated position the error persist; on the contrary, with the feedback this error is disappeared, as seen previously in Bode diagram.



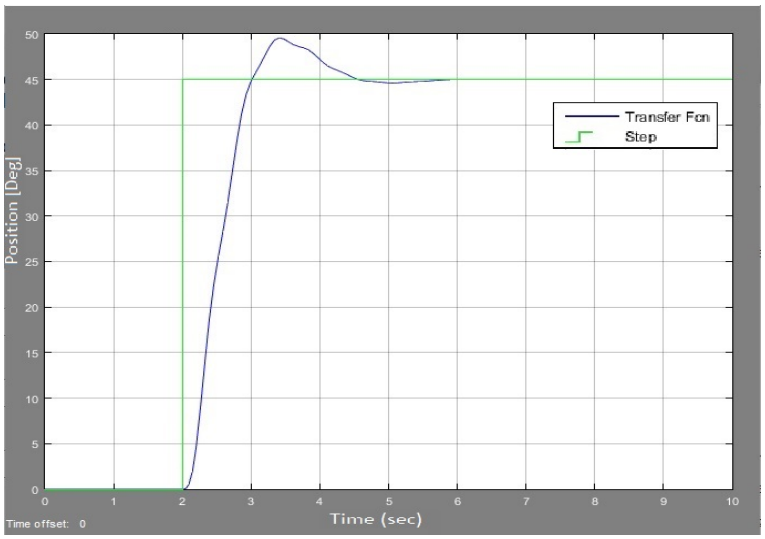


Figure 5.9: Simulation result of joint angle θ_j without and with feedback of τ_{env}, respectively

At the end, experiment is made to see the response of the real system. Results are visible in the figures 5.10:

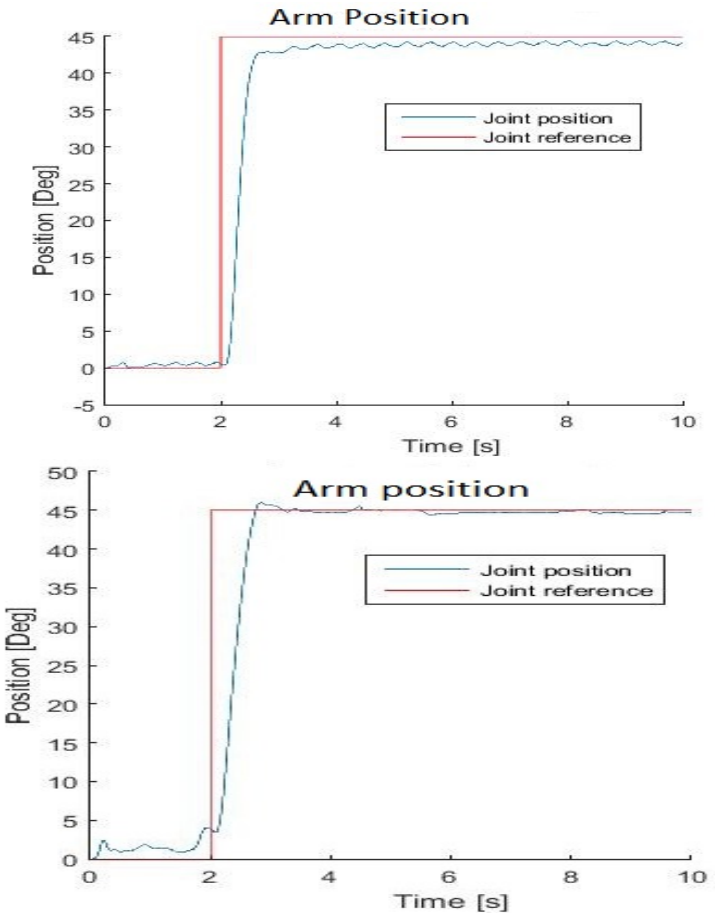


Figure 5.10: System's response of joint angle θ_j without and with feedback of τ_{env}

Observing these two last figures, by implementing the parameters found earlier, the system provides a better response. In fact, whereas previously the system couldn't reach reference's value and its response was with continuous oscillations; now it follows almost perfectly the reference with less changes. This means that the implementation is correct because there is a compensation of the disturbance from the surrounding environment, as described previously by Bode diagrams, supported by simulations and confirmed by experiment.

Chapter 6

On-line Identification

6.1 Adaptive Control

In everyday language, "to adapt" means to change a behaviour to conform to new circumstances. Intuitively, an *adaptive control* is a controller that can modify its behaviour in response to changes in the dynamics of the process and the character of the disturbances. Then:

```
an adaptive controller is a controller with adjustable parameters
and a mechanism for adjusting these parameters.
```

Listing 6.1: Definition of adaptive controller

The controller becomes non-linear because of the parameter adjustment mechanism. It has, however a special structure. Since it is difficult to deal with general non-linear systems, it makes sense to consider special classes of non-linear systems.
An adaptive control system can be though of as having two loops:

- one loop is a normal *feedback* with the process and the controller;
- the other loop is the *parameter adjustment* loop.

A block diagram of an adaptive system is shown in figure 6.1. The parameter adjustment loop is often slower than the normal feedback loop.
A control engineer should know about the potential of the adaptive systems, because they have useful properties, which can be profitably used to design control system with improved performance and functionality.

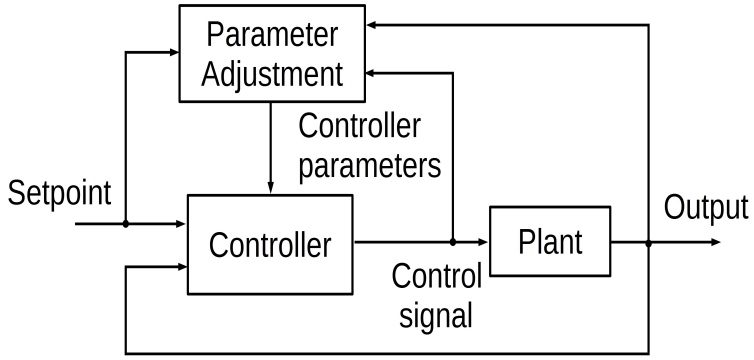


Figure 6.1: Block diagram of an adaptive system

6.1.1 Adaptive Schemes

There are some adaptive systems that are used according to the desired specifications to obtain.

- *Gain Scheduling*:

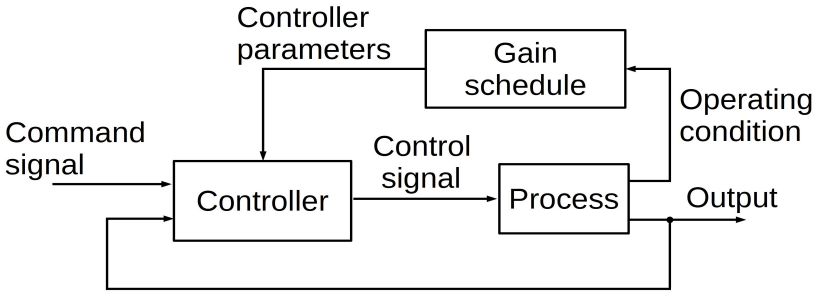


Figure 6.2: Block diagram of a system with gain scheduling

sometimes it is possible to find measurable variables that correlate well with changes in process dynamics. These variables can then be used to change the controller parameters. This approach is called "gain scheduling" because the scheme was originally used to measure the gain

and then change, that is scheduled by the controller to compensate the changes. The concept of gain scheduling originated in connection with the development of flight control systems. In this application the Mach number and the altitude are measured by air data sensors and used as scheduling variables.

- *Model-Reference Adaptive Systems (MRAS)*:

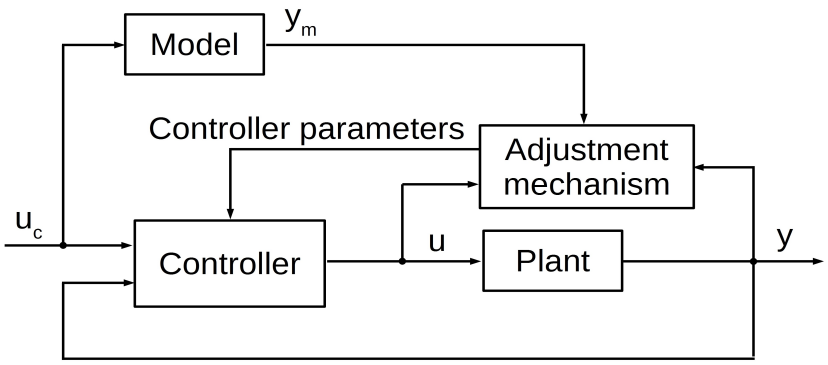


Figure 6.3: Block diagram of a model-reference adaptive system (MRAS)

the MRAS tells how the process output ideally should respond to the command signal. The controller can be thought of as consisting of two loops. The inner loop is an ordinary feedback loop composed of the process and the controller. The outer loop adjust the controller parameters in such a way that the error, which is the difference between process output y and model output y_m, is small. The MRAS was originally introduced for flight control. In this case the reference model describes the desired response of the aircraft to joystick motions.
The key problem with MRAS is to determine the adjustment mechanism so that a stable system, which brings the error to zero, is obtained. The following parameter adjustment mechanism, called the *MIT rule*, was used in the original MRAS

$$\frac{d\theta}{dt} = -\gamma e \frac{\delta e}{\delta \theta}$$

In this equation, $e = y - y_m$ denotes the model error and θ is a controller parameter. The quantity $\delta e/\delta\theta$ is the sensitivity derivative of the error with respect to parameter *theta*. The parameter γ determines the adaptation rate.
In practice it is necessary to make approximations to obtain the sensitivity derivative. The MIT rule can be regarded as a gradient scheme to minimize the squared error e^2.

- *Self-tuning Regulators (STR)*:

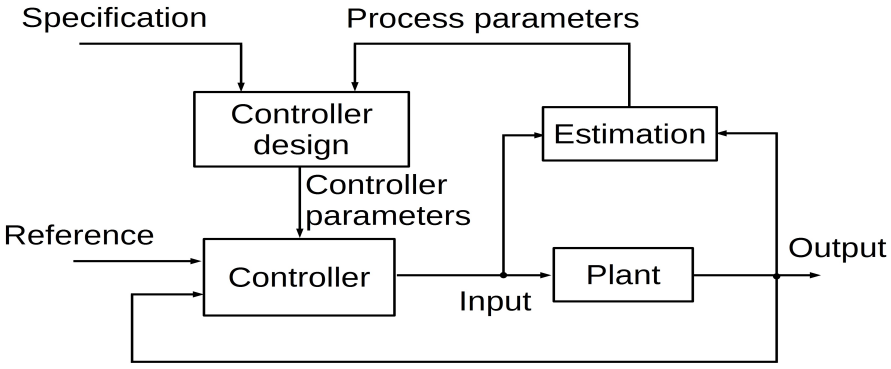


Figure 6.4: Block diagram of a self-tuning regulator (STR)

the adaptive schemes discussed so far are called *direct* methods, because the adjustment rules tell directly how the controller parameters should be updated.
A different schemes is obtained if the estimates of the process parameters are updated and the controller parameters are obtained from the solution of a design problem using the estimated parameters. A block diagram of such a system is shown in figure 6.4. The adaptive controller can be thought of as being composed of two loops. The inner loop consists of the process and an ordinary feedback controller. The parameters of the controller are adjusted by the outer loop, which is composed of a recursive parameter estimator and a design calculation. Sometimes, it is not possible to estimate the process parameters without introducing probing control signals or perturbations. Notice that the system may be viewed as an automation of process modelling and design, in which the process model and the control design are update at each sampling period. A controller of this construction is called *self-tuning regulator (STR)* to emphasize that the controller automatically tunes its parameters to obtain the desired properties of the closed-loop system.
The block diagram in figure 6.4 represents an on-line solution to a design problem for a system with known parameters. This is the *underlying design problem*. Such a problem can be associated with most adaptive control schemes, it is often useful to find the underlying design problem, because it will give the characteristics of the system under the ideal conditions when the parameters are exactly known.
The STR scheme is very flexible with respect to the choice of the un-

derlying design and estimation methods. Many different combinations have been explored. The controller parameters are updated indirectly via the design calculations in the self-tuner shown in figure 6.4. It is sometimes possible to re-parametrize the process so that the model can be expressed in terms of the controller parameters. This gives a significant simplification of the algorithm because the design calculations are eliminated. In terms of figure 6.4 the controller parameters are updated directly.
In the STR the controller parameters of the process parameters are estimated in real time. The estimates are then used as if they were equal to the true parameters (i.e., the uncertainties of the estimates are not considered). This is called the *certainty equivalence principle*. In many estimation schemes it is also possible to get a measure of the quality of the estimates. This uncertainty may then be used in the design of the controller. For example, if there is a large uncertainty, one may choose a conservative design.

6.1.2 Recursive-Least-Square algorithm (RLS)

On-line determination of process parameters is a key element in adaptive control. A recursive parameter estimator appears explicitly as a component of a self-tuning regulator in figure 6.4. Parameter estimation also occurs implicitly in a model-reference adaptive controller.
In adaptive controllers, the observation are obtained sequentially in real time. It is then desirable to make the computations recursively to save computation time. Computation of the least-squares estimate can be arranged in such a way that the results obtained at time $t-1$ can be used to get the estimates at time t.
Let $\alpha(t-1)$ denote the least-squares estimate based on $t-1$ measurements, the estimate at time t can be written as:

$$\hat{\alpha}(t) = \hat{\alpha}(t-1) + k(t)[y(t) - \gamma^T(t)\hat{\alpha}(t-1)] \tag{6.1}$$

where

$$\begin{aligned} k(t) &= P(t)\gamma(t) = \\ &= P(t-1)\gamma(t)[\beta I + \gamma(t)^T P(t-1)\gamma(t)]^{-1} \end{aligned} \tag{6.2}$$

$$P(t) = \frac{1}{\beta}[I - k(t-1)\gamma^T(t)]P(t-1) \tag{6.3}$$

$$\varepsilon(t) = y(t) - \gamma^T(t)\hat{\alpha}(t-1) \tag{6.4}$$

The residual $\varepsilon(t)$ can be interpreted as the error in predicting the signal $y(t)$ one step ahead based on the estimated $\hat{\alpha}(t-1)$.
Equation (6.1) has strong intuitive appeal. The estimate $\hat{\alpha}(t)$ is obtained by adding a correction to the previous estimate $\hat{\alpha}(t-1)$. The correction is proportional to $y(t)-\gamma^T(t)\hat{\alpha}(t-1)$. The correction term is thus proportional to the difference betwen the measured value of $y(t)$ and the prediction of $y(t)$ based on the previous parameter estimate. The components of the vector $k(t)$ are weighting factors that tell how the correction and the previous estimate should be combined.[26]
These equations provides the estimate $\hat{\alpha}$ that minimizes the function cost:

$$J(\alpha,t)=\frac{1}{2}\sum_{i=1}^{t}\beta^{t-i}[y(t)-\gamma^T(i)\alpha(t)]^2 \tag{6.5}$$

where $0<\beta\leq 1$ is called *forgetting factor*, as said in Chapter 5.

6.2 On-line environment's identification in tendon arm system

In a tendon arm mechanism, some considerations are to be made in order to obtain the best results for the case under investigation.

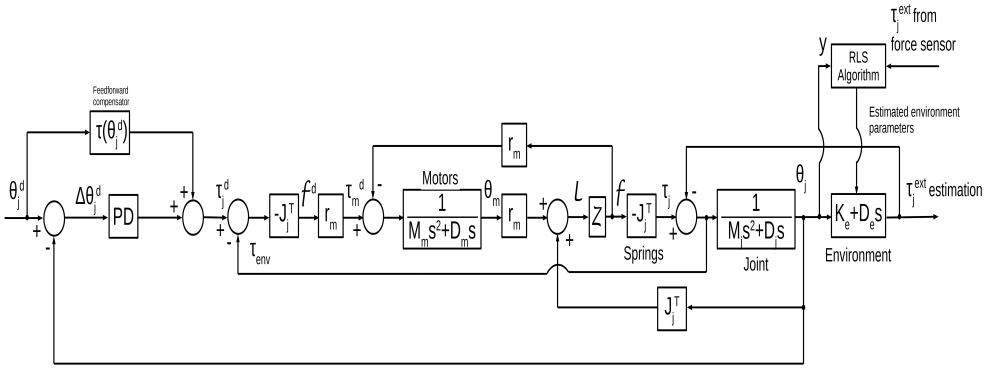


Figure 6.5: Block diagram for the on-line identification

6.2.1 Equivalence of the block diagram and RLS calculations

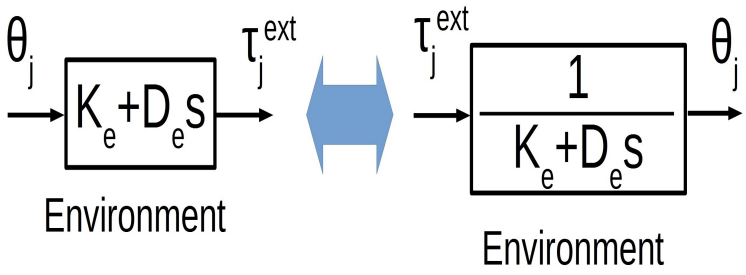


Figure 6.6: Block diagram for the on-line identification

As can be seen, the transfer function concerning the environment is composed by one zero and no pole, which makes impossible to find the corresponding z-transform in order to obtain the filtered data. To solve the problem, only in RLS identification block, input and output are inverted. In fact, according to the general rules of block diagrams, the two representations in figure 6.6 are equivalent.

In this way it's possible to write the transfer function in the following way, without committing any mistake:

$$\frac{1}{Ke + Des} = \frac{1}{De}\frac{1}{s+a} \tag{6.6}$$

where $a = Ke/De$. The corresponding z-transform is:

$$\frac{1}{De}\frac{z}{z - e^{-aT}} \tag{6.7}$$

The RLS estimation is performed on the filtered data:

$$u_f = H_u(z)u \tag{6.8}$$

$$y_f = H_y(z)y \tag{6.9}$$

with

$$H_u(z) = D_e z \tag{6.10}$$

$$H_y(z) = \frac{z^2}{z - e^{-aT}} \tag{6.11}$$

The structure of the model to identify is therefore:

$P'(z) = \dfrac{Y_f(z)}{U_f(z)} = \dfrac{H_y(z)}{H_u(z)}\dfrac{Y}{U} = \dfrac{b_0 z}{z - a_0}$, with $a_0 = e^{-aT}$ and $b_0 = 1/De$.

The regression form is:

$y_f(t) - y_f(t-1)a_0 = u_f(t)b_0 \;\Rightarrow y_f(t) = y_f(t-1)a_0 + u_f(t)b_0 =$

$$[y_f(t-1)\; u_f(t)] \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \gamma^T(t)\alpha(t).$$

Then in the equations (6.1), (6.2), (6.3) of the RLS algorithm:[27]

$$\hat{\alpha}(t) = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} e^{-aT} \\ 1/De \end{bmatrix}, \; y(t) = \Theta_j(t) = \text{joint's position and}$$

$$\gamma(t) = \begin{bmatrix} y_f(t-1) \\ u_f(t) \end{bmatrix}.$$

The $\beta's$ value is imposed to 0.9999 in order to keep in memory the previous values.
The initial value P_0 of the covariance matrix $P(t)$ is chosen as:

$P_0 = 0.001 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, small because, by the simulation and experiment made before, there is the certainty that the previously estimated parameters are correct.
At the end, the initial value of α is:

$\alpha_0 = \begin{bmatrix} e^{-aT} \\ 1/De \end{bmatrix}$, where $a = \dfrac{ke}{De} = \dfrac{3.7819}{3.8482}$, which are the estimated values find by the off-line identification method with MATLAB toolbox, and T=sampling time=0.001 sec.

6.2.2 Conditional updating

A problem, called *estimator windup*, arises in RLS with exponential forgetting when the input excitation is poor. Two "extreme" situations can be considered:

1. There is no excitation, i.e.
   $\gamma(t) = 0$. The RLS step becomes $\hat{\alpha}(t) = \hat{\alpha}(t-1)$,
   $P(t) = P(t-1)/\beta \Longrightarrow \hat{\alpha}(t)$ remains constant, while $P(t)$ grows unbounded. Since the RLS gain is $k(t) = P(t)\gamma(t)$, the estimate $\hat{\alpha}(t)$ may change dramatically when $\gamma(t)$ starts to change again.

2. $\gamma(t)$ spans only a subset $\Re^m$ of the parameter space $\Re^n$ $(n > m)$. It may happen that $\hat{\alpha}(t)$ grows unbounded, but its "projection" $\gamma^T(t)\hat{\alpha}(t)$ remains constant. Then, if output $y(t)$ is constant, the innovation:
   $\varepsilon(t) = y(t) - \gamma^T(t)\hat{\alpha}(t-1)$
   in (6.4) is also constant, so that the RLS procedure doesn't have to perform any correction to stop the growth of the estimate.

Estimator windup can be avoided by adopting a *conditional updating* scheme for RLS estimate. This scheme consists of updating the RLS estimate $\alpha(t)$ and covariance matrix $P(t)$ only when the plant is "sufficiently excited". Common tests for detecting a "sufficient excitation" condition are based on monitoring the magnitudes of variations of process inputs and outputs, or other signals such as ε (innovation) or $\gamma^T P\gamma$.
The conditional updating scheme is:

$$\gamma^T(t)P(t)\gamma(t) > 2(1-\beta) \tag{6.12}$$

if the condition is verified then the excitation is sufficiently "rich" and the RLS estimate can be updated.
If it's not verified the previous data of $\alpha(t)$ and $P(t)$ are loaded.[27]

6.2.3 White noise and Lagged-Fibonacci generator

Another point that must be taken into account is when data is generated under "noisy" conditions, i.e.

$$y(t) = \gamma^T(t)\alpha_0 + e(t) \tag{6.13}$$

then the RLS method provides a "correct" estimate $\hat{\alpha}$ of the "true" parameters α_0, only if $e(t)$ is a *white noise*, that is also independent of $\gamma(t)$. "Correct" estimate means that $\hat{\alpha}$ is:

- *unbiased*, i.e. $E[\hat{\alpha} = \alpha_0]$, with $E[.]$ denotes the expectation operator;
- *consistent*, i.e. $\hat{\alpha} \Rightarrow \alpha_0$ as $t \to +\infty$.[27]

A white noise is defined as a *random* vector "w" having its average value and its autocorrelation matrix:
$\mu_w = E[w] = 0$,
$R_{ww} = E[ww^T] = \sigma^2 I$.
White noise, thus, has its instantaneous values entirely without correlation. Its sample signal is composed by values completely unpredictable in respect to the previous, so with a high content of information. In fact, by looking at figure 6.7, it's spectrum is almost continuous.

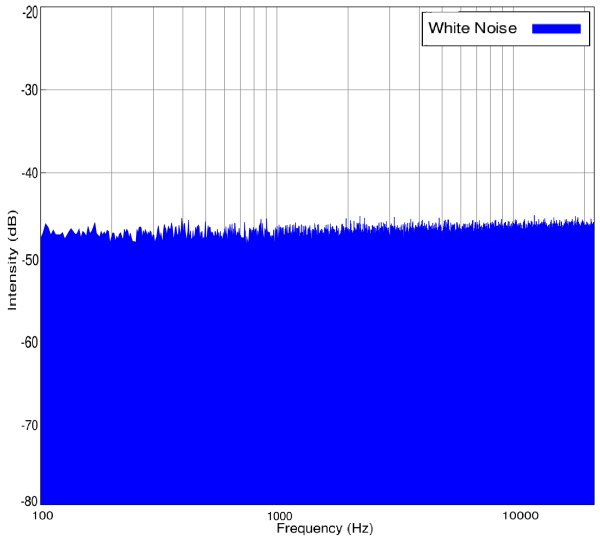


Figure 6.7: Example of white noise spectrum

To generate white noise, the simple *random* instruction in C program, is not enough (figures 6.8 will clarify the motivation, and it's explanation will be

exposed later on).
The best solution is the *Lagged Fibonacci Generator*, this algorithm is used for the generation of pseudo-random numbers based on a generalization of the *Fibonacci's Sequence*. The Fibonacci's generator is defined as:
$F_n = (F_{n-j} + F_{n-k}) mod(m)$,
$0 < j < k$, where:

- F_n is the n-th term of the numbers sequence generated using *random* instruction;
- F_{n-j} and F_{n-k} are any two previous terms of the sequence;
- $mod(m)$ indicates the rest of the division between $(F_{n-j} + F_{n-k})$ and (m).

6.2.4 Bartlett test

As said previously, the spectral content of white noise is characterized by a uniform spectrum in the considered band. To verify the spectral content of the output y(t), *"Bartlett Test"* is used.
Consider $\{y(t)\}_{t=1,..,N}$ sample of the process $\{y(t)\}_{t\in Z}$, the estimation of $e(t)$ is:
$\hat{S}_e(\theta_k) = |E(\theta_k)|^2$, where $\theta_k = \dfrac{2\Pi}{N}k$ with $k = 0, .., N-1$, and

$$E(\theta_k) = \frac{1}{\sqrt{N}} \sum_{h=0}^{N} y(h) e^{-j\theta_k h} \tag{6.14}$$

By observing equation (6.14) Bartlett test (or *periodogram cumulated*, *whiteness test*) execute a cumulative sum of the module of the *"Fast Fourier Transform-FFT"* of the signal under test. The closer *innovation* is near to the *theoretical value*, the better will be the spectral content, and therefore, the on-line parameters estimation will be improved.

Figures 6.8 show "Bartlett test" respectively without "Lagged Fibonacci Generator" thus with only "random" instruction and with implementation of the "Lagged Fibonacci Generator". In the first figure innovation is far from the theoretical value; while in the second one, innovation is more closer.

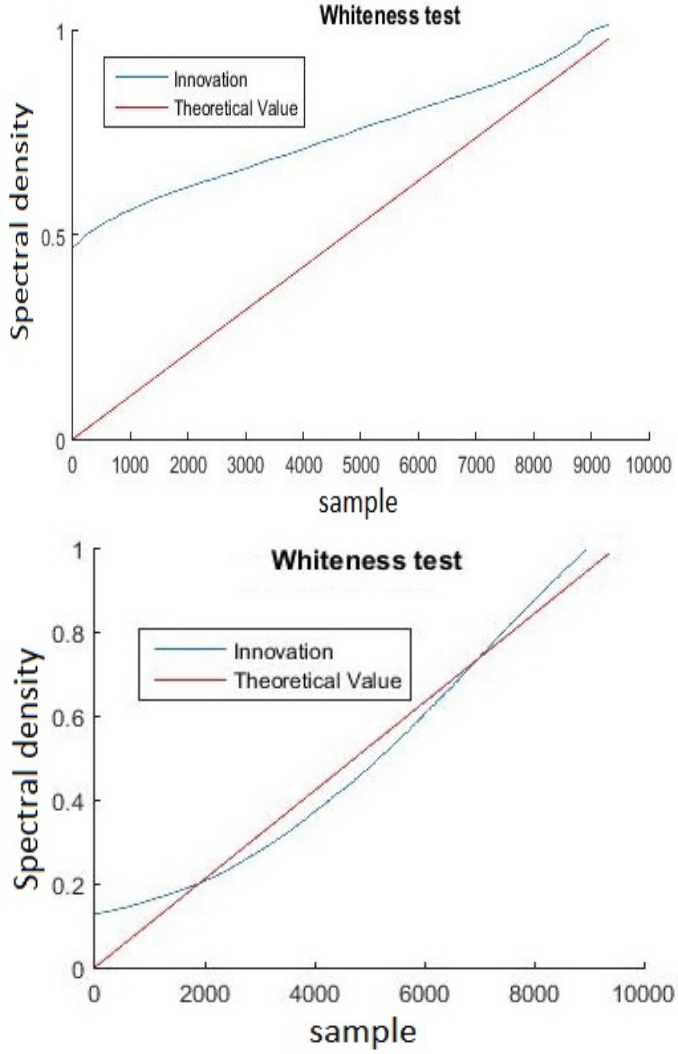


Figure 6.8: Bartlett test without and with "Lagged Fibonacci Generator", respectively

6.2.5 Experiment of the on-line identification

In this part the theory explained earlier is put into practice.
Tests are made to see the on-line estimation's accuracy of the parameters and how the system responds with them.
Figure 6.9 shows the joint's position in a time of 15 seconds divided:

- *0-0.5 seconds*: parameters identified in off-line mode are kept in memory;
- *0.5-10 seconds*: on-line identification of the environment's parameters with the "Recursive-Least-Square" algorithm;
- *10-15 seconds*: parameters identified at tenth second are keep in memory to verify system's response, this last part has been enlarged in figure 6.12.

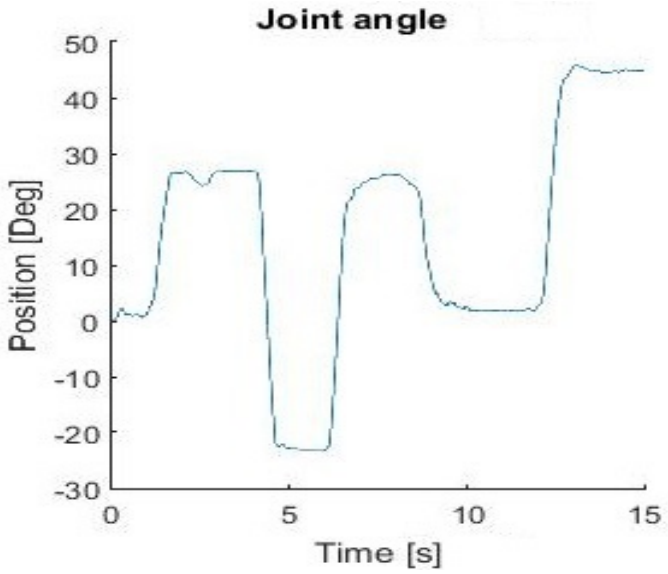


Figure 6.9: Joint's position from 0 to 15 seconds

In Figures below, parameters change their value in order to *adapt* the system to the new environment characteristic. For this reason they are slightly different from the reference parameters found with the MATLAB identification toolbox explained in Chapter 5; this means that the implementation made is correct.

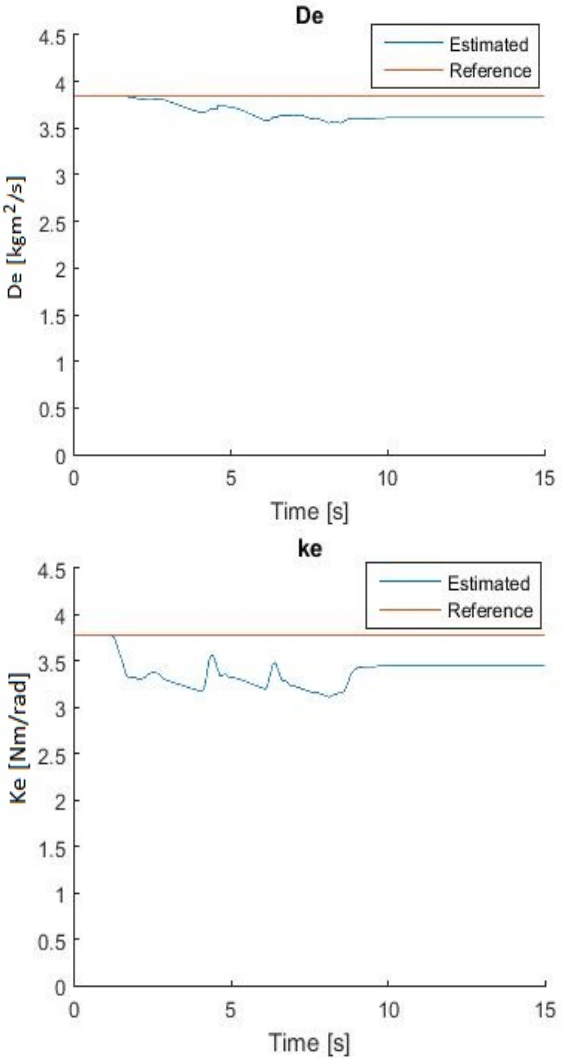


Figure 6.10: *De* and *ke* parameters, respectively

In order to see the system's response, after the on-line part, which goes from the tenth and fifteenth second, parameters "K_e" and "D_e" are kept constants and equal as the last value found on the tenth second.

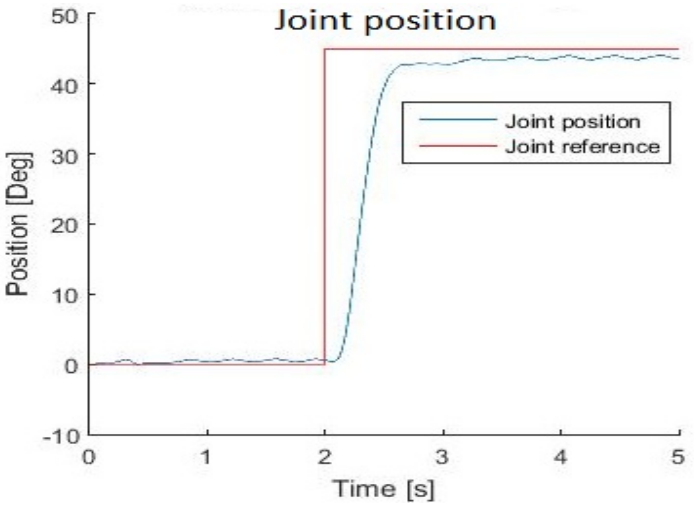


Figure 6.11: Joint's position with conventional method

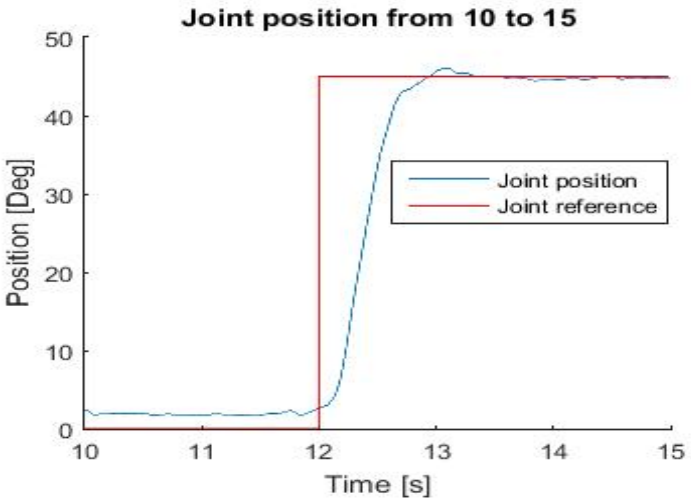


Figure 6.12: Joint's position with proposed method

In figures 6.11, 6.12 are shown reference position and real position of the joint without on-line identification and with on-line identification, respectively. In particular in the second one, where identified parameters by on-line method are keep in memory and equal to $k_e = 3.4490$ and $D_e = 3.6120$, real position follows well reference trajectory with less oscillations.
At the end, figures 6.13, 6.14 show comparison between measured external torque and estimated external torque. The first one is found by the force sensor placed at the end of the joint, while the second one is calculated by estimated on-line parameters multiplied to the joint position. The figures show that both external torque have similar trends. So by observing figures from 6.11 to 6.14, the part relative to the on-line identification of environment's parameters works successfully.

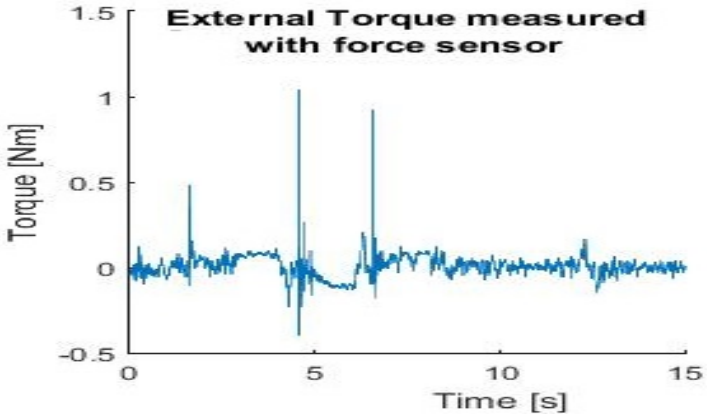


Figure 6.13: Measured external torque

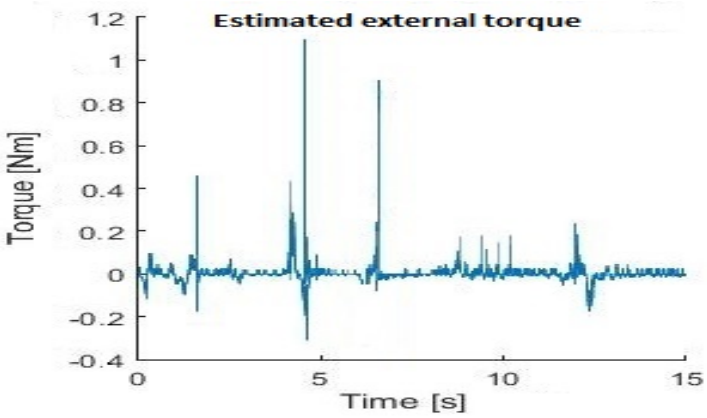


Figure 6.14: Estimated external torque

Chapter 7

Conclusions

In the first part of this thesis, the off-line identification was carried out using MATLAB toolbox to estimate the environment's parameters "K_e" and "D_e" in which tendon arm is placed. Always with the aim of verifying this method, "Recursive Least Square" is used in off-line mode, which confirmed that the first method is right. After having obtained the estimated parameters, a simulation was made in order to see the response of the simulated system, in particular the simulated stiffness and position of the tendon arm. Simulation shows that there is an improvement of the system.
In the second part, experiment was executed to see the real response of the system. By watching Bode diagrams, responses of the transfer function and response of the real tendon arm, figures 5.10, show that there is a better tracking of the reference, which confirms the improvement of the system showed by simulation.
The final part is relative to the on-line identification of the environment's parameters. For the purpose of estimating these parameters a "Recursive-Least-Square" algorithm is used, in this case in real time. To implement this part some considerations were made. The equivalence of the block diagram is used in order to find the relative Z-transform (6.7) to obtain the filtered data (6.8), (6.9). The second implementation is the wind-up estimator with the conditional updating in order to load in the program the estimated vector $\alpha(t)$ and the covariance matrix $P(t)$ ((6.1), (6.3) of RLS, respectively), only when the excitation (6.12) is "sufficiently rich". At the end, a white noise is implemented to obtain a good estimation. By using "Bartlett test" in figures 6.8 it was shown that in order to generate a white noise, a simple "random" instruction is not sufficient; therefore, a "Lagged Fibonacci Generator" is implemented.
Figures 6.10 show that reference and estimated parameters have similar trends. Figures 6.11, 6.12 set out the joint position in the conventional

method and proposed method in steady state condition, respectively; in the second figure, reference and real position are mostly overlapping. At the end figures 6.13, 6.14 show the external torque measured with the force sensor and the estimated external torque found by using the estimated parameters. Trends of both external torques are almost the same, confirming thereby the correctness of the on-line identification.

Appendix A

Lagrange equations of the tendon arm

Define:

- K = Kinetic Energy;
- U = Potential Energy;
- L = Lagrangian Function = $K - U$.

By observing figure 4.3, the Lagrange equation is:

$$\frac{d}{dt}(\frac{\delta L}{\delta \dot{q_i}}) - \frac{\delta L}{\delta q_i} = u_i \quad (i = 1, 2, .., n) \tag{A.1}$$

In particular the kinetic energy is defined $K = K_1 + K_2$:

$$\begin{aligned} K_1 &= \frac{1}{2} m_1 \dot{x_{g1}}^2 + \frac{1}{2} m_1 \dot{y_{g1}}^2 + \frac{1}{2} I_1 \dot{\theta_1}^2 \\ K_2 &= \frac{1}{2} m_2 \dot{x_{g2}}^2 + \frac{1}{2} m_2 \dot{y_{g2}}^2 + \frac{1}{2} I_2 (\dot{\theta_1} + \dot{\theta_2})^2 \end{aligned} \tag{A.2}$$

with $m_{1,2}$ the masses of the two link, $I_{1,2}$ the inertia of the joint and

$$\begin{aligned}
x_{g1} &= l_{g1}\cos\theta_1 \\
y_{g1} &= l_{g1}\sin\theta_1 \\
x_{g2} &= l_1\cos\theta_1 + l_{g2}\cos(\theta_1) + \theta_2 \\
y_{g2} &= l_1\sin\theta_1 + l_{g2}\sin(\theta_1) + \theta_2 \\
&\Downarrow d/dt \\
\dot{x_{g1}} &= -l_{g1}\dot{\theta_1}\sin\theta_1 \\
\dot{y_{g1}} &= l_{g1}\dot{\theta_1}\cos\theta_1 \\
\dot{x_{g2}} &= -l_1\dot{\theta_1}\sin\theta_1 - l_{g2}(\dot{\theta_1}+\dot{\theta_2})\sin(\theta_1+\theta_2) \\
\dot{y_{g2}} &= l_1\dot{\theta_1}\cos\theta_1 + l_{g2}(\dot{\theta_1}+\dot{\theta_2})\cos(\theta_1+\theta_2)
\end{aligned} \tag{A.3}$$

Then, the equations (A.2) can be rewrite:

$$\begin{aligned}
K_1 &= \frac{1}{2}m_1(-l_{g1}\dot{\theta_1}\sin\theta_1)^2 + \frac{1}{2}m_1(l_{g1}\dot{\theta_1}\cos\theta_1)^2 + \frac{1}{2}I_1\dot{\theta_1}^2 \\
&= \frac{1}{2}(m_1 l_{g1}^2 + I_1)\dot{\theta_1}^2 \\
K_2 &= \frac{1}{2}m_2\{-l_1\dot{\theta_1}\sin\theta_1 - l_{g2}(\dot{\theta_1}+\dot{\theta_2})\sin(\theta_1+\theta_2)\}^2 \\
&+ \frac{1}{2}m_2\{l_1\dot{\theta_1}\cos\theta_1 + l_{g2}(\dot{\theta_1}+\dot{\theta_2})\cos(\theta_1+\theta_2)\}^2 + \\
&+ \frac{1}{2}I_2(\dot{\theta_1}+\dot{\theta_2})^2 = \\
&= \frac{1}{2}\{m_2 l_1^2\dot{\theta_1}^2 + 2m_2 l_1 l_{g2}\dot{\theta_1}(\dot{\theta_1}+\dot{\theta_2})\cos\theta_2 + (m_2 l_{g2}^2 + I_2)(\dot{\theta_1}+\dot{\theta_2})^2\}
\end{aligned} \tag{A.4}$$

The potential energy is defined as: $U = U_1 + U_2$:

$$\begin{aligned}
U_1 &= m_1 g l_{g1}\sin\theta_1 \\
U_2 &= m_2 g\{l_1\sin\theta_1 + l_{g2}\sin(\theta_1+\theta_2)\}
\end{aligned} \tag{A.5}$$

The vector containing the free coordinates is:

$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$; and finally the generalized forces:

$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$ where τ_i is the torque $[Nm]$. Thus:

$$\begin{aligned}
\frac{\delta K_1}{\delta \dot{\theta_1}} &= (m_1 l_{g1}^2 + I_1)\dot{\theta_1} \\
\frac{\delta K_2}{\delta \dot{\theta_1}} &= m_2 l_1^2 \dot{\theta_1} + m_2 l_1 l_{g2}(\dot{\theta_1} + \dot{\theta_2})\cos\theta_2 + m_2 l_1 l_{g2}\dot{\theta_1}\cos\theta_2 + (m_2 l_{g2}^2 + I_2)(\dot{\theta_1} + \dot{\theta_2}) \\
&= m_2 l_1^2 \dot{\theta_1} + m_2 l_1 l_{g2}(2\dot{\theta_1} + \dot{\theta_2})\cos\theta_2 + (m_2 l_{g2}^2 + I_2)(\dot{\theta_1} + \dot{\theta_2})
\end{aligned} \tag{A.6}$$

by deriving them other time:

$$\begin{aligned}
\frac{d}{dt}(\frac{\delta K_1}{\delta \dot{\theta_1}}) &= (m_1 l_{g1}^2 + I_1)\ddot{\theta_1} \\
\frac{d}{dt}(\frac{\delta K_2}{\delta \dot{\theta_1}}) &= m_2 l_1^2 \ddot{\theta_1} + m_2 l_1 l_{g2}(2\ddot{\theta_1} + \ddot{\theta_2})\cos\theta_2 - m_2 l_1 l_{g2}(2\dot{\theta_1} + \dot{\theta_2})\dot{\theta_2}\sin\theta_2 + \\
&\quad + (m_2 l_{g2}^2 + I_2)(\ddot{\theta_1} + \ddot{\theta_2}) = \\
&\quad = (m_2 l_1^2 + m_2 l_{g2}^2 + I_2 + 2m_2 l_1 l_{g2}\cos\theta_2)\ddot{\theta_1} + (m_2 l_{g2}^2 + I_2 + m_2 l_1 l_{g2}\cos\theta_2)\ddot{\theta_2} \\
&\quad - m_2 l_1 l_{g2}(2\dot{\theta_1} + \dot{\theta_2})\dot{\theta_2}\sin\theta_2
\end{aligned} \tag{A.7}$$

The potential energy is without $\dot{\theta_1}$, then
$\frac{\delta U}{\delta \dot{\theta_1}} = 0;$
$\frac{d}{dt}(\frac{\delta U}{\delta \dot{\theta_1}}) = 0.$

The first term of the Lagrange equation (A.1) is then:

$$\begin{aligned}
\frac{d}{dt}(\frac{\delta L}{\delta \dot{\theta_1}}) &= \frac{d}{dt}(\frac{\delta K_1}{\delta \dot{\theta_1}}) - \frac{d}{dt}(\frac{\delta K_2}{\delta \dot{\theta_1}}) \\
&= \frac{d}{dt}(\frac{\delta K_1}{\delta \dot{\theta_1}}) + \frac{d}{dt}(\frac{\delta K_2}{\delta \dot{\theta_1}}) = (m_1 l_{g1}^2 + m_2 l_1^2 + m_2 l_{g2}^2 + I_1 + I_2 + 2m_2 l_1 l_{g2}\cos\theta_2)\ddot{\theta_1} \\
&+ (m_2 l_{g2}^2 + I_2 + m2 l_1 lg2\cos\theta_2)\ddot{\theta_2} - m_2 l_1 l_{g2}(2\dot{\theta_1} + \dot{\theta_2})\dot{\theta_2}\sin\theta_2
\end{aligned} \tag{A.8}$$

In the kinetic energy there isn't the term θ_1 so: $\dfrac{\delta K}{\delta \theta_1} = 0$; the potential energy instead is:

$$\frac{\delta U_1}{\delta \theta_1} = m_1 g l_{g1} \cos\theta_1$$
$$\frac{\delta U_2}{\delta \theta_1} = m_2 g \{ l_1 \cos\theta_1 + l_{g2} \cos(\theta_1 + \theta_2) \}$$

The second term of the Lagrange equation is:

$$\begin{aligned} \frac{\delta L}{\delta \theta_1} &= \frac{\delta K}{\delta \theta_1} - \frac{\delta U}{\delta \theta_1} = -\frac{\delta U_1}{\delta \theta_1} - \frac{\delta U_2}{\delta \theta_1} \\ &= -(m_1 g l_{g1} + m_2 g l_1) \cos\theta_1 - m_2 g l g 2 \cos(\theta_1 + \theta_2) \end{aligned} \tag{A.9}$$

By composing the equations (A.8) and (A.9), the Lagrange equation can be rewrite:

$$\begin{aligned} \tau_1 &= (m_1 l_{g1}^2 + m_2 l_1^2 + m_2 l_{g2}^2 + I_1 + I_2 + 2 m_2 l_1 l_{g2} \cos\theta_2) \ddot{\theta_1} + \\ &+ (m_2 l_{g2}^2 + I_2 + m_2 l_1 l g 2 \cos\theta_2) \ddot{\theta_2} - m_2 l_1 l_{g2} (2\dot{\theta_1} + \dot{\theta_2}) \dot{\theta_2} \sin\theta_2 + \\ &+ (m_1 g l_{g1} + m_2 g l_1) \cos\theta_1 + m_2 g l_{g2} \cos(\theta_1 + \theta_2) \\ \tau_2 &= (m_2 l_{g2}^2 + I_2 + m_2 l_1 l_{g2} \cos\theta_2) \ddot{\theta_1} + \\ &+ (m_2 l_{g2}^2 + I_2) \ddot{\theta_2} + m_2 l_1 l_{g2} \dot{\theta_1}^2 \sin\theta_2 + m_2 g l_{g2} \cos(\theta_1 + \theta_2) \end{aligned} \tag{A.10}$$

Appendix B

MATLAB code of the simulations

```
%%%%%%%%%% Flags setup %%%%%%%%%%%%%%%%%%%%%%%%%%
sp.experswi = 4;
sp.satswi   = 2;
sp.gswi     = 0;
sp.noise    = 1;
sp.wm       = 0;
sp.fsensor  = 0;
sp.Cswi     = 0;

%%%%%%%%%% Sample time and Initial value %%%%%%%
sp.samptime = 0.001;  %[sec]
sp.trqlimit = 3;      %[Nm]
sp.flimargin= 0;      %[N]
sp.Fref     = 0.3;    %[N]
sp.Fthr     = 1;      %[N]
sp.pitch    = 4;      %[s]
sp.preq(1)  = -90;    %[deg]
sp.preq(2)  = 10;     %[deg]
sp.qmargin  = 0.08;   %[rad]

%%%%%%%%%% Filter value %%%%%%%%%%%%%%%%%%%%%%%%%
cp.DOBcut   = 500;    %[2LPF]
cp.mTC      = 400;    %[3LPF]
cp.Grcut    = 300;    %[3LPF]
cp.fCcut    = 150;    %[4LPF]
cp.RFOBcut  = 100;    %[2LPF]
cp.qTC      = 10;     %[3LPF]
cp.WRFOBcut = 200;
cp.Sfilcut  = 500;
cp.frefcut  = 100;
cp.qrefcut  = 5;
cp.srefcut  = 20;
```

```
%%%%%%%%%% Joint's parameters %%%%%%%%%%%%%%%%%%
cp.ma = [0.375 0.260];
cp.Ga = [0.1185 0.100];
cp.La = [0.2455 0.200];
cp.Ia = [0.0051 0.004];
cp.Ra = [0.017 0.015];
cp.wa = cp.Ra*2;
cp.Ja = cp.ma*(diag(cp.La)^2+diag(cp.wa)^2)/12;

%%%%%%%%%% Motor's parameters %%%%%%%%%%%%%%%%%%
cp.rp = diag([0.055   0.055   0.055   0.055   0.055   0.055]);
cp.Dp = diag([0.00448 0.00350 0.00403 0.00374 0.00498 0.00351]);
cp.Jp = diag([0.00581 0.00453 0.00505 0.00363 0.00463 0.00398]);

%%%%%%%%%% SAT parameters %%%%%%%%%%%%%%%%%%%%%%%
cp.SATap = [  5.0   5.2   5.5   5.7   5.7   5.1    ;
              187   192   182   179   186   185 ]' ;

%%%%%%%%%% Jacobian matrix %%%%%%%%%%%%%%%%%%%%%%
cp.Jj=[cp.Ra(1) -cp.Ra(1) cp.Ra(1) -cp.Ra(1) cp.Ra(1) -cp.Ra(1)
       0         0         cp.Ra(2) -cp.Ra(2) -cp.Ra(2) cp.Ra(2)]';

[sp.mdof sp.adof]=size(cp.Jj);
sp.snum=sp.adof*(sp.adof+1)/2;

%%%%%%%%%% Equations 4.5 %%%%%%%%%%%%%%%%%%%%%%%
if(sp.adof==1)
    cp.Mn = cp.Ja(1) + cp.Ga(1)^2 * cp.ma(1);
    cp.In = cp.Ia(1);
elseif(sp.adof==2)
    cp.Mn(1,1) = cp.Ja(1)+cp.Ja(2)+cp.ma(1)*cp.Ga(1)^2+
               + cp.ma(2)*(cp.La(1)^2+cp.Ga(2)^2)+
               + 2*cp.ma(2)*cp.La(1)*cp.Ga(2)/2^(1/2);
    cp.Mn(1,2) = cp.Ja(2)+cp.ma(2)*cp.Ga(2)^2+
               + cp.ma(2)*cp.La(1)*cp.Ga(2)/2^(1/2);
    cp.Mn(2,1) = cp.Mn(1,2);
    cp.Mn(2,2) = cp.Ja(2)+cp.ma(2)*cp.Ga(2)^2;
    cp.In = diag(cp.Ia);
else end

%%%%%%%%%% Motors matrices %%%%%%%%%%%%%%%%%%%%%%
for t=1:sp.mdof
    cp.r(t,t)=cp.rp(t,t);
    cp.D(t,t)=cp.Dp(t,t);
    cp.J(t,t)=cp.Jp(t,t);
end
cp.inv_r=inv(cp.r);
```

```
cp.SATa=cp.SATap(1:sp.mdof,:);
mp.preq=sp.preq'/180*pi;
mp.pretheta=-inv(cp.r)*cp.Jj*mp.preq;
mp.prexy(1)= cp.La(1) * cos(mp.preq(1)) +
       + cp.La(2) * cos(mp.preq(1)+mp.preq(2));
mp.prexy(2)= cp.La(1) * sin(mp.preq(1)) +
       + cp.La(2) * sin(mp.preq(1)+mp.preq(2));
mp.qref=zeros(sp.adof,1);
mp.qref(1,1)=1;
mp.fref=ones(sp.mdof,1);

%%%%%%%%%% Force threshold %%%%%%%%%%%%%%%%%%%%%%%%
cp.fmax=sp.trqlimit/cp.r(1,1)-sp.flimargin;
cp.fmin=sp.flimargin;

%%%%%%%%%% Equation 4.13 %%%%%%%%%%%%%%%%%%%%%%%%%%
cp.i(1)=1;
cp.i(2)=1;
for t=1:sp.snum
    cp.H(t,:)=cp.Jj(:,cp.i(2))'*diag(cp.Jj(:,cp.i(1)));
    if(cp.i(1)==sp.adof)
        cp.i(2)=cp.i(2)+1;
        cp.i(1)=cp.i(2);
    else
        cp.i(1)=cp.i(1)+1;
    end;
end
cp.C=diag(cp.SATa(:,2)');
cp.HC=cp.H*cp.C;
cp.JaT=[-cp.Jj';cp.HC];
cp.pinvJaT=pinv(cp.JaT);
cp.Hd=cp.H*diag(-cp.SATa(:,1))*(-cp.SATa(:,2));
cp.medisj=cp.H*cp.C*cp.fmax*ones(sp.mdof,1)/2+cp.Hd;

mp.ma = cp.ma*(1-0.05*randn*sp.wm);
mp.Ga = cp.Ga;
mp.La = cp.La;
mp.Ja = cp.Ja*(1-0.05*randn*sp.wm);
mp.Ia = cp.Ia*(1-0.05*randn*sp.wm);
mp.Ra = cp.Ra;
mp.Jj = cp.Jj;

mp.r = cp.r;
mp.D = cp.D*(1-0.05*randn*sp.wm);
mp.J = cp.J*(1-0.05*randn*sp.wm);

%%%%%%%%%% SAT values %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mp.SAT  = [ 5.0  5.2  5.5  5.6  5.7  5.1
            185  190  180  177  184  183
```

```
            5.0  5.1  4.8  5.1  5.0  4.9 ]';

mp.time=0.2;
if(sp.satswi==2)
  mp.lfa=600;
else
    mp.lfa=0;
end;
if(sp.satswi==0)
  mp.SAT(:,1:2)=cp.SATa;
end;
mp.pre=zeros(sp.mdof,1);
for t=1:sp.mdof
    mp.motor(t,t)=tf(1,[mp.J(t,t) mp.D(t,t)]);
end
for t=1:sp.adof
    cp.WRFOB_fil(t,t)=tf(cp.WRFOBcut,[1 cp.WRFOBcut])^3;
    cp.Sfil(t,t)=tf(cp.Sfilcut,[1 cp.Sfilcut])^2;
end

%%%%%%%%%% Filter transfer function %%%%%%%%%%%%%%%
cp.DOB_fil=tf(cp.DOBcut,[1 cp.DOBcut])^2;
cp.RFOB_fil=tf(cp.RFOBcut,[1 cp.RFOBcut])^2;
cp.qreffil=tf(cp.qrefcut,[1 cp.qrefcut])^4;
cp.sreffil=tf(cp.srefcut,[1 cp.srefcut])^4;
cp.freffil=tf(cp.frefcut,[1 cp.frefcut])^4;
cp.ftaucut=50;
cp.fraufil=tf(cp.ftaucut,[1 cp.ftaucut])^2;

%%%%%%%%%% Motors transfer function %%%%%%%%%%%%%%%
zp.motor=c2d( mp.motor , sp.samptime , 'tustin');

%%%%%%%%%% Disturbance observer  transfer function %
zp.DOB_fil=c2d( cp.DOB_fil , sp.samptime , 'tustin');
zp.RFOB_fil=c2d( cp.RFOB_fil , sp.samptime , 'tustin');
zp.WRFOB_fil=c2d(cp.WRFOB_fil,sp.samptime,'tustin');
zp.Sfil=c2d(cp.Sfil,sp.samptime,'tustin');
zp.qreffil=c2d(cp.qreffil,sp.samptime,'tustin');
zp.sreffil=c2d(cp.sreffil,sp.samptime,'tustin');
zp.freffil=c2d(cp.freffil,sp.samptime,'tustin');
zp.ftaufil=c2d(cp.fraufil,sp.samptime,'tustin');

%%%%%%%%%% PD controller values %%%%%%%%%%%%%%%%%%%%
for t=1:sp.mdof
    cp.mcut(t,t)=3*cp.mTC  -cp.D(t,t)/cp.J(t,t);
    cp.Kd(t,t)   =3*cp.mTC^2*cp.J(t,t)/cp.mcut(t,t);
    cp.Kp(t,t)   =  cp.mTC^3*cp.J(t,t)/cp.mcut(t,t);
    cp.mfil(t,t) = tf(cp.mcut(t,t),[1 cp.mcut(t,t)]);
    cp.Gm(t,t)   = tf(cp.mTC,[1 cp.mTC])^3 ;
```

```
end
zp.mfil=c2d(cp.mfil,sp.samptime,'tustin');
zp.Gm=c2d( cp.Gm , sp.samptime , 'tustin');
cp.Gr=tf(cp.Grcut,[1 cp.Grcut])^3;
zp.Gr=c2d( cp.Gr , sp.samptime , 'tustin');
cp.fC=tf(cp.fCcut,[1 cp.fCcut])^4;
zp.fC=c2d( cp.fC , sp.samptime , 'tustin');
for t=1:sp.adof
    cp.qcut(t) = 3*cp.qTC - cp.Ia(t)/cp.Mn(t,t);
    cp.qKd(t)  = 3*cp.Mn(t,t)*cp.qTC^2/cp.qcut(t) - cp.Ia(t);
    cp.qKp(t)  = cp.Mn(t,t)*cp.qTC^3/cp.qcut(t);
    cp.qcon(t,t)=tf([cp.qKd(t) cp.qKp(t)],1)
                *tf(cp.qcut(t),[1 cp.qcut(t)]);
end
zp.qcon=c2d(cp.qcon,sp.samptime,'tustin');

cp.FKi=0;
mp.Me=1000000;            %[kg]
mp.De=1000000;            %[kg/s]
mp.xye0=[0.2  -0.4];    %[m]

%%%%%%%%%% Off-line environment's parameters %%%%%%
load('tf4_1.mat');
%tf1=c2d(tf1,t)
[numEnv,denEnv]=tfdata(tf4_1,'v');
mp.Dae=abs(numEnv(2));    %[kg/s]
mp.Kae=abs(numEnv(1));    %[N/m]
for t=1:2
    mp.object(t,t)=tf(1,[mp.Me mp.De]);
end;

zp.object=c2d(mp.object,sp.samptime,'tustin');
sp.xy_marg=sp.Fref/mp.Kae;

for t=1:sp.snum
  sp.srand(t)=t+sp.adof;
end;
for t=1:sp.adof
  sp.qnoice(t)=t+sp.adof+sp.snum;
end;
for t=1:sp.mdof
  sp.mnoice(t)=t+sp.adof+sp.snum+sp.adof;
end;

%%%%%%%%%% With/without disturbance observer %%%%%
sp.Cswi=0; %interruttore
run main_exp_box\main_exp_4

figxyC=figxy;
```

```
figsC =figs;
figFC =figF;
dataSC =dataS;

sp.Cswi=1;
run main_exp_box\main_exp_4

drawdata
```

Listing B.1: MATLAB code

Bibliography

[1] SICILIANO, KHATIB, B. -O., *Springer Handbook of Robotics*, Springer-Verlag Berlin Heidelberg, 2008.

[2] Passenberg-Groten-Peer-Buss, C.-R.-A.-M, *Towards Rea-Time Haptic Assistance Adaptation Optimizing Task Performance and Human Effort,* Proc. IEEE World Haptics Conference, p.155-160, 2011.

[3] Sawers-Ting, A.-L., *Perspectives on human-human sensorimotor interactions for the design of rehabilitation robots,* Journal of neuroengineering and rehabilitation, 2014.

[4] *https://www.en.wikipedia.org/wiki/Industrial-robot.*

[5] *https://www.osha.gov/dts/osta/otm/otm-iv/otm4-iv-4.html.*

[6] MILNER, T., *Muscle-tendon mechanical impedance. http://www.sfu.ca/tmilner/kin416.htm,2006*, School of Kinesiology, Simon Fraser University.

[7] ZATSIORSKY, V., *kinetics of Human Motion. No. ISBN: 0736037780.* humankinetics.com, 2002.

[8] HAIYA-KOMADA-HIRAI, K.-S.-H., *Tension Control for Tendon Mechanisms by Compensation of Nonlinear Spring Characteristic Equation Error.*, Mechatronic Department, MIE University.

[9] Pratt-Williamson, G.-M., *Series elastic actuators,* Proc. IEEE/RSJ Int. Conf. Intelligent Robots and System, vol. 1, p.339, 1995.

[10] De Luca-Siciliano-Zollo, A.-B.-L., *PD control with on-line gravity compensation for robots with elastic joints: Theory and experiment,* Automatica, vol. 41, no. 10, p.1809-1819, 2005.

[11] Rost-Verl, A.-A., *The quadhelix-drive - an improved rope actuator for robotic aplications,* in Proc. IEEE Int. Conf. Robotics and Automation, p. 3254-3259, 2010.

[12] Ishikawa-Komi-Grey-Lepola-Bruggemann, M.-P.V.-M.J.-V.-G.P., *Muscle-tendon interaction and elastic energy usage in human walking,* Journal of applied physiology 99, 2005.

[13] Williamson, -M.M., *Series Elastic Actuators* M.Eng., University of Oxford, Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science, Massachusetts Institute of technology, February, 1995.

[14] Laffranchi-Tsagarakis-Cannella-Caldwell, M.-N.G.-F.-V.-D.G., *Antagonistic and Series Elastic Actuators: a Comparative Analysis on the Energy Consumption* IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, October 11-15, 2009.

[15] Bicchi-Tonietti, -A.-G., *Fast and "Soft-Arm" Tactics,* Dealing with the Safety-Performance Tradeoff in Robot Arms Design and Control.

[16] Arumugom-Muthuraman-Ponselvan, S.-S.-V., *Modeling and Application of Series Elastic Actuators for Force Control Multi Legged Robots. https://sites.google.com/site/journalofcomputing, ISSN: 2151-9617*, Journal of computing, Volume 1, Issue 1, December 2009.

[17] Villani-De Schutter, L.-J. *Force Control*, Chapter 7, pag.161-164.

[18] Khatib, O., *Force-Based Motion Control of Robot Manipulators*, Robotics Laboratory, Computer Science Department, Stanford University, Stanford, California, USA.

[19] Colgate, Hogan E.-N., *The interaction of robots with passive environments: applications to force feedback control.*, In Advanced Robotics 1989, Proceedings of the Fourth International Conference on International Robotics, pages 465-474, Columbus, Ohio, June 1989.

[20] Eppinger, Seering S.D.-W.P., *On Dynamic Models of Robot Force Control*, In Proceedings of the IEEE International Conference on Robotics and Automation, pages 392-397, 1989.

[21] Paljug, Sugar, Vijay Kumar, Yun E.-T.G.-R.-X, *Important Consideration in Force Control With Applications to Multi-Arm Manipulation*,

Technical Reports (CIS), University of Pennsylvania, Department of Computer & Information Science, October 1991.

[22] Zinn, Khatib, Roth, Kenneth Salisbury M.-O.-B.-J, *A New Actuation Concept for Human-Friendly Robot Design Playing It Safe*, IEEE Robotics and Automation Magazine, pag. 502-507, 2004.

[23] Shirai, Tanaka, Tomiok T.-N.-T., *Mechanism and Characteristic of Non-Linear Spring SAT*, Robotics Society of Japan Scholarly Lecture Presentation '03, 2003. (in Japanese)

[24] Komada, Inadama, Ohnishi S.-S.-K., *Bilateral Robot Hand Based on Estimated Force Feedback*, in Proc. IEEE IECON'87, Vol.2, pp.602-607, 1987.

[25] Komada, Hirai, Kageyama, Yashiro S.-J.-T.-D., *A Study on Mechanical Stiffness in Force Control of Tendon Arm*, in IEEJ International Workshop on Sensing Actuation, Motion Control, and Optimization, 2016.

[26] Astrom, Wittenmark K.,J. -B., *Adaptive control, Second Edition*, Dover publications, Mineola, New York, 2008.

[27] Oboe R., *Control System Design, Self Tuning Regulator for a DC servomotor*, Dept. of Management and Engineering University of Padova (Vicenza branch), April 2016.