Obrero: A platform for sensitive manipulation

Eduardo Torres-Jara
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: etorresj@csail.mit.edu

***Abstract*— We are interested in developing *sensitive manipulation* for humanoid robots: manipulation that is as much about perception as action and is intrinsically responsive to the properties of the object being manipulated; manipulation that does not rely on vision as the main sensor but as a complement. Therefore, we have developed a platform that consider the requirements need it.**

Humans are capable of manipulating objects in a dexterous way in unstructured environments. We use our limbs not only as pure actuators but also as active sensors. Human manipulation is so sensitive that many tasks can be accomplished using our hands without help from vision. In contrast, humanoid robots in general are limited in the operations they can perform with their limbs.

In order to achieve *sensitive manipulation*, we plan to use a behavior-based architecture to deal with unknown environments. Traditionally, the trajectory of the robotic manipulator is completely planned based on a model of the world (usually a CAD model). This renders the manipulator incapable of operating in a changing environment (not to mention an unknown one) unless a model of the environment is acquired in real-time.

The same situation was faced by mobile robotics where behavior-based architecture was introduced as an alternative to the traditional method which relies on a model of the world. Behavior-based architecture has proven successful in mobile robots operating in unstructured and dynamic environments. Consequently, we want to take advantage of the features of behavior-based control for achieving sensitive manipulation.

In addition to the control architecture, a platform for sensitive manipulation needs: force control, dense tactile sensing and contact compliance. In order to grab or move an object, we need to sense and control the force applied by the end effector. Dense tactile sensing allows us to detect edges, texture, motion, and other properties of the objects. This information is used to determine the manipulator's next action. The third requirement deals with the limb coming in contact with an object.

When contact occurs the platform needs to respond fast enough to avoid damaging itself or the object. In practice, when the limb comes in contact with an object the passive elements of the system are the ones that determine the response. Therefore, these passive elements must have a low mechanical impedance to achieve contact compliance. This property is especially important when using the limb as an active exploring device.

Consequently, to achieve sensitive manipulation an adequate platform must first be implemented.

In this paper, we present the design, construction and evaluation of a platform adequate for sensitive manipulation. The platform consists of a force controlled arm, a sensitive hand, and an active vision head. It uses non-conventional actuators, high density tactile sensors, force control and low mechanical impedance.

We start by designing and characterizing a series elastic actuator for the hand. This actuator is used to design and build a 5 degrees of freedom, force controlled hand with dense tactile sensing and low mechanical impedance. We also describe the arm used and the design of the vision system. All these components are interconnected by a high speed, low overhead communication network. We conclude by presenting a tentative behavior-based control schematic for a specific task.

I. Introduction

We are interested in developing *sensitive manipulation* for humanoid robots: manipulation that is as much about perception as action and is intrinsically responsive to the properties of the object being manipulated; manipulation that does not rely on vision as the main sensor but as a complement.

In this paper, we present the design, construction and evaluation of a humanoid platform (Obrero) suitable for sensitive manipulation. The design of the platform is motivated by human manipulation. Humans are capable of manipulating objects in a dexterous way in unstructured environments. We use our limbs not only as pure actuators but also as active sensors. Human manipulation is so sensitive that many tasks can be accomplished using our hands without any help from vision. In contrast, humanoid robots in general are limited in the operations they can perform with their limbs alone.

However, if we consider tasks such as precise positioning or accurate repeated motion of an arm, we notice that, in general, humans are outperformed by robots because human limbs are clumsier than robotic ones. This apparent disadvantage is overcome by the great number of sensors and actuators present in human limbs which allows us to adapt to different conditions of the environment.

For instance, humans use their hands to touch or grab an object without damaging themselves or the object. This is possible because humans can control the force and the mechanical impedance exerted by their limbs when in contact with an object. Robots, in general, cannot do this because their components lack the sensing and actuating capabilities needed to control these parameters (i.e., the force and the impedance).

Motivated by these ideas, we have favored the sensing capabilities over the precision in the design of Obrero's limb. The limb has force control, low mechanical impedance as well as position and force sensing.

Moreover, the sensing capabilities of human limbs are not limited to force. Humans can also extract many features of an object they are holding [**?**] thanks to their highly innervated skin. In contrast, robotic limbs have a limited number of sensors, rendering them inadequate for feature extraction.

The great sensitivity of the human limbs makes manipulation quite independent from other sensory modalities such as vision. As an example, consider the scenario in which we are looking for a TV remote control on a coffee table in a dark room. A person can move her hand on top of the table until she hits the remote (assuming there is no other object on the table). Then she can move her hand around the object to identify a familiar shape, such as that of a button, and consequently conclude that she found the remote. The complete task can be executed thanks to the information provided by sensors located in the hand and arm that permit exploring the environment and identifying the remote without damage.

In the platform that we present, we address the mechanical and perceptual requirements of such actions. We use non-conventional actuators for the hand and arm and dense tactile sensors for the hand (special attention is paid on the actuators in the hand because of size constrains). These actuators control the force, reduce the mechanical impedance, and protect the motors against shocks. These features allow the limb to come in contact with objects in a safe manner. For instance, when contact occurs the platform needs to respond fast enough to avoid damaging itself or the object. In practice, when the limb comes in contact with an object the passive elements of the system are the ones that determine the response. Therefore, these passive elements must have a low mechanical impedance to achieve contact compliance. This property is especially important when using the limb as an active exploring device.

While tactile information will dominate, *sensitive manipulation* also can benefit from visual and auditory perception. Such information will be used by the robot to improve the efficiency of manipulation, rather than be an essential prerequisite. Vision can give a quick estimate of an object's boundary or find interesting inhomogeneities to probe. Sound is also a very important clue used by humans to estimate the position of an object and to identify it [**?**].

The robot Obrero has a 2 degree-of-freedom head that includes vision and sound. The camera has two optical degrees of freedom; focus and zoom. Focus is very useful to obtain depth information and zoom helps to obtain fine details of an image. The vision system will try to take advantage of natural cues present in the environment such as shadows [**?**].

In order to achieve *sensitive manipulation*, we plan to use a behavior-based architecture to deal with unknown environments. Traditionally, the trajectory of the robotic manipulator is completely planned based on a model of the world (usually a CAD model). This renders the manipulator incapable of operating in a changing environment (not to mention an unknown one) unless a model of the environment is acquired in real-time.

The same situation was faced by mobile robotics where behavior-based architecture was introduced as an alternative to the traditional method which relies on a model of the world. Behavior-based architecture has proven successful in mobile robots operating in unstructured and dynamic environments. Consequently, we want to take advantage of the features of behavior-based control for achieving sensitive manipulation.

In the following sections we present the platform Obrero. We start by designing and characterizing a series elastic actuator for the hand. This actuator is used within the design of a 3 fingered, force controlled hand with dense tactile sensing and low mechanical impedance. We also describe the 6 DOF force controlled arm ([**?**]) used and the design of the head.

In section II we include background literature in areas relevant to the topic. We end with conclusions in section **??**.

mds
November 18, 2002

II. Background Literature

A. Tactile Sensing

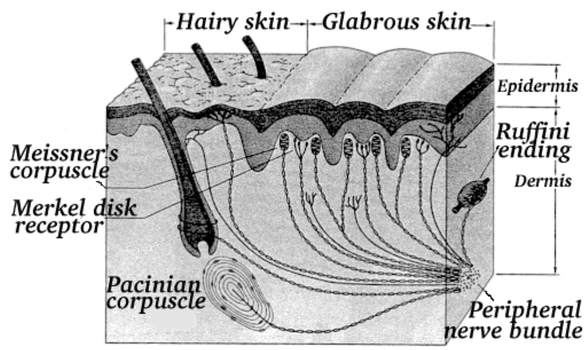

Fig. 1. Mechanoreceptors present in hairless (glabrous) and hairy human skin.

1) Human Body: The human body is completely covered by a sensitive skin which allows us to get information from the environment. The information comes in four modalities [**?**]: discriminative touch, proprioception, nociception and temperature.

These modalities supply the following kinds of information:

- Discriminative touch. Shape, size, texture and movement of an object.
- Proprioception. Position and movement of the limbs and body.
- Nociception. Tissue damage.
- Temperature. Perception of cold, cool, warm and hot.

All this information is collected by mechanoreceptors which are different in the hairless and hairy skin as can be observed in figure 1. These mechanoreceptors are classified as superficial and deep receptors.

The superficial receptors are known as Meissner's corpuscles and Merkel's disks. Meissner's corpuscles are rapidly adapting sensors that have fine mechanical sensitivity. On the other hand Merkel disks are slowly adapting sensors that measure strain.

The deep receptors are larger than the superficial receptors. They are known as Pacinian corpuscules and Ruffini endings. The Pacinian corpuscules respond to rapid indentations of the skin but not to steady pressure. They also detect vibration.

Ruffini endings are slow adapting sensors that link subcutaneous tissue to foldings in the skin in places such as the joints and finger nails.

In the hairy skin there are similar receptors. There are two different types: the hair follicle receptor and the field receptor. They detect hair movement and skin stretches respectively. Both receptors are rapid adapting ones.

The information collected by a group of mechanoreceptors is collected by a dorsal root ganglion neuron. This group of receptors only represents a small area of skin and is known as a receptive field.

Receptive fields for different mechanoreceptors are different. For example, the receptive fields of Meissner corpuscules or Merkel's disk are composed of 10-25 mechanoreceptors. In the case of Meissner corpuscules, this corresponds to circular receptive fields with diameters of 2-3mm in the finger's tip and 10mm in the palm. The size of the receptive fields makes these mechanoreceptors ideal for detecting fine features in an object. On the other hand receptive fields of the Ruffini endings and Pacinian corpuscules cover a wider area, which makes them good at detecting coarse features.

All these corpuscules are distributed in different densities in the skin. This gives different spatial resolution in different parts of the body. The hand is one of the richest areas regarding tactile sensing.

The information from these mechanoreceptors is sent to the somatosensory cortex. The signals go through several relay regions. When the signals arrive at the cortex, the cortical neurons have larger receptive fields than the ones of the dorsal root ganglion neurons. That allows the detection of more complex features.

All the somatosensory inputs that arrive at the cortex from a somatosensory map known as the homunculus. The cortex itself is organized in columns about 300-600 μm wide where each column corresponds to only one location and modality. The space occupied in the homunculus by the different inputs is not proportional to its physical size but to its density of innervation. However, the spaces occupied by the different parts of the body are not fixed and they can be changed by experience.

The patterns detected by the mechanosensors are faithfully reproduced in the cortex maps up to the first stage of the cortical map in area 3b [**?**]. In higher cortical levels, neurons are activated by specific combination of receptive fields. This allows detecting specific features of objects. For example, researchers have identified neurons that are sensitive to: orientation, direction, texture and shape.

2) Robotics: In robotic systems, tactile sensing is not as rich as in humans. This is basically because the technology to create dense sensing is not available. However, many attempts have been made to implement tactile sensing in robots. There are many technologies used to build sensor arrays: conductive elastomers, elastomer-dielectric capacitive, optical sensors (surface motion and frustrated internal reflection), piezoelectric, acoustic, magnetoelastic, electromagnetic dipoles, silicon micromechanical (mems), and force sensing resistors. A complete review of these technologies can be found in [**?**].

The performance of these sensors has been measured according to the parameters mentioned in a survey study by Harmon [**?**]. Those parameters include: spatial and temporal resolution, measurement accuracy, noise rejection, hysteresis, linearity, number of wires, packing, and cost. However, it is not clear if many of these designs are useful for manipulation because little attention has been given to the data produced by these sensors [**?**].

Another approach for sensing tactile forces has been to use joint torque and force information to recover the normal forces [**?**] instead of using superficial sensors. Nevertheless, this approach is only able to detect resulting forces as opposed to distributions.

B. Platforms for Robotic Manipulation

In robotics, several researchers have designed and constructed arms with different features depending on the application to address. For example we can mention: Milacron's arm, PUMA 560, WAN [**?**], DLR arm [**?**] and Cardea's arm [**?**]. The same applies to the design of hands where we can mention: the MIT/Utah's [**?**], the Stanford/JPL [**?**], the Barret's [**?**], DLR's [**?**] and the Shadow's [**?**] hand. There is also a wealth of work in the area of wrists. However, there are only a few platforms that have been constructed to research manipulation as a whole. Not surprisingly most of these platforms are humanoid robots. In this section, we will pay attention to these platforms.

- **Dexter** is a humanoid platform which has two Whole Arm Manipulators (WAM) [**?**], two Barrett hands [**?**], and a BiSight stereo head.
  The WAM arms have 7 direct-drive DOFs and cable transmission for force control. Each of the hands has 3 fingers and 4 DOFs. One DOF for each finger and one for rotating the fingers. The tips of each finger have an ATI load cell for force sensing. The BiSight stereo head can pan, tilt, and independently verge each camera. The cameras have control of focus, iris, and zoom. The head also has a binaural acoustic sensor consisting of four microphones. A VME architecture is used for computation.
  This platform has capabilities for exploring its environment using the compliance of the arms. However, the hands are not compliant. The work implemented in this platform [**?**], [**?**] shows an extensive use of force sensing in the fingers to deal with objects of unknown geometry. The speed of operation is limited, in part because of the lack of compliance in the fingers.
- **Robonaut** is a humanoid robotic platform designed to operate in space. It consists of a 2 DOF head (pan/tilt) and stereo cameras, two 7 DOF arms with force/torque cells at each shoulder (16 embedded sensors at each DOF), and two 14 DOF hands [**?**] whose design is based on the MIT-UTAH hand. The tactile sensing is still in development (miniature force cells for the fingertips)

but currently uses FSRs. This robot was designed to manipulate the same kinds of tools that humans do in space, controlled by teleoperators. However, due to the time-delays in communication the platform is becoming more autonomous.
Autonomous and semi-autonomous manipulation uses force-sensing from a few force cells in the shoulder and wrist. The arms are designed for high stiffness and consequently the harmonic geardrives are prone to damage. To solve this, the robot is covered with padding. Therefore, this platform is not fully designed to conduct exploration with its limbs.
- **Cog** is a humanoid robot designed to study embodied intelligence and social interaction. Cog has twenty-two mechanical DOFs: two 6 DOF arms, a 3 DOF torso, a 4 DOF neck, and 3-DOF combined in its eyes. The actuators in the arm are series elastic actuators [**?**]. Its design allows the robot interact safely with its environment and with people. These capability have been exploited in [**?**], [**?**] and [**?**].
- **Saika** is a humanoid robot [**?**] that consists of a two-DOF neck, dual five-DOF upper arms, a torso and a head. The hands and forearms used were designed according to the tasks to perform. The control used was behavior-based. Some of the goals of the robot were: hitting a bouncing ball, grasping unknown objects and catching a ball [**?**].

C. *Grasping and manipulation*

Extensive work has been done in the area of grasping objects using robotic hands. A very detailed summary of the field can be found in [**?**]. An extensive modelling and analysis of different aspects of the mechanics of grasping has been developed. Those aspects include kinematics of the hands, modelling of contact, stability of grasping, robustness of grasp, and dynamics of the hand-object system.

Salisbury [**?**] has presented an analysis of the kinematics and forces in a hand. He starts by classifying the contact points between fingers and objects. The contact points can be modelled as: frictionless, frictional, or soft. A finger with a frictionless contact only exerts force on the direction of the normal to the object. If the contact is frictional, the finger exerts normal and tangential forces. In the case of a soft contact, the finger exerts a torsional torque in addition to the normal and tangential forces. Once the contact points are modelled the conditions of a stable grasping are defined. This equilibrium is known as force closure. A more constrained definition of stable grasping is known as form closure. Under form closure, the object grasped can resist external disturbances. A more extensive analysis of force and form closure can be found in [**?**].

Further modelling of contact points includes the analysis of kinematics and compliance. The kinematics of a contact point as two bodies move has been derived by [**?**] for planar bodies and by [**?**] for rigid spatial bodies with extensions by [**?**]. Compliance of the contacts has also been modelled in robotics; representative work is presented in [**?**],[**?**],[**?**]. However, the modelling of compliance has proven to be a more difficult task than the modelling of kinematics (See [**?**]).

Another aspect that has been considered in manipulation is the robustness of a grasp, which is defined as the ability to reject small disturbances from external forces and/or torques. The assumption made in this measure is that the fingers can be positioned accurately. Although this gives some idea of the performance of grasping, a better theory is needed.

In addition, the dynamics of the hand-object system has been analyzed. These analyses include the control laws of the hand/fingers and are mainly done in simulation. A major problem found is the inconsistency obtained when dynamics of the rigid bodies and the dynamics of the contacts are used together. Such is the case of “peg-in-the-hole” analysis by [**?**] and [**?**]. They showed cases where either no or two solutions for the accelerations were found. The inconsistency is attributed to the contact model used, i.e., Coulomb model. Therefore, efforts have been made to use a compliant contact model instead [**?**]. However, the analysis in simulation is still hard because of the great difference in time scale between the dynamics of contacts and that of rigid bodies.

All the work presented up to this point relies on the existence of models for both the object to manipulate and the fingers. Recently, there have been new approaches to deal with objects of unknown geometry. These approaches rely more on the information provided by the sensors than on the pre-existent model.

For example, [**?**] treats the problem of grasping as a controller composition problem. This is similar to behavior-based architectures used in mobile robotics, where simple behaviors are combined to accomplish a task. In [**?**], it is assumed that the fingers are in contact with the object being grasped. The controllers move the fingers iteratively until they get to a stable position. The controllers are combined in a hierarchical manner, using a nullspace projection to determine the region (in space state) in which the controllers do not interfere with each other.

An aspect of manipulation that has been mostly neglected is exploration. That is using the manipulator to learn from the world as opposed to blindly acting on it. This has been mainly because of the way manipulators have been built. A starting point to this approach is [**?**] where a force controlled arm was used to explore its environment. The information obtained from this exploration was used to improve objet segmentation.

There is also a great amount of work in task and trajectory planning for manipulation, we will not review these topics because our approach is a different one. The reader is referred to [**?**].

III. Robot Obrero

In this section we present the design and implementation of the robot Obrero. The overall architecture of the robot is presented in section III-A. The design and implementation of the hand and its actuation system are described in sections III-C and III-B. The robotic arm and head are described in sections III-D and III-E. Finally, the software architecture is described in section III-F.

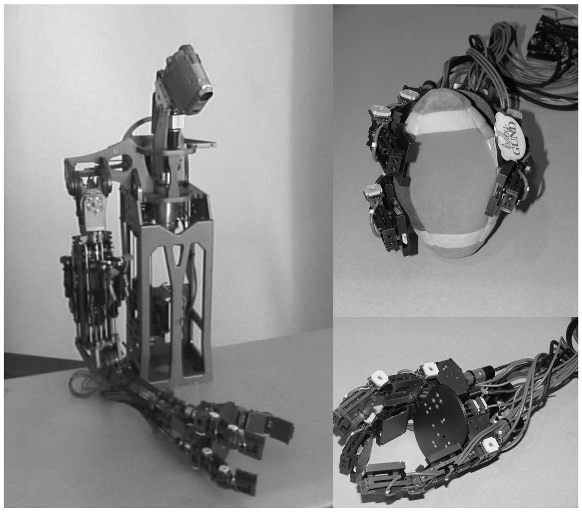

Fig. 2. Robot Obrero. The picture shows the head, arm and hand of the robot. In the upper-right corner we can observe the hand grabbing a ball.

A. Robot Hardware Architecture

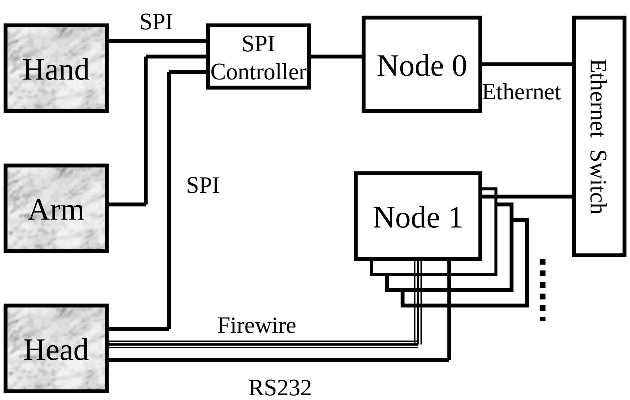

Fig. 3. Overall architecture of Obrero. The motor controllers of the Hand, Arm and Head are connected to a linux node via a SPI communication module. The head is also connected to rest of the linux network via firewire for acquiring images/sound and via RS-232 to control zoom and focus.

The robot Obrero is shown in figure 2 where we can observe the hand, arm, torso and head. Obrero's overall hardware architecture is presented in figure 3. In this latter figure we can observe that the hand, arm and head controllers connect to a communication board with three SPI channels (5Mbps). The communication board interfaces with a EPP parallel port in a linux computer. The details about the hand, arm and head controllers are explained in sections III-C.4, III-D and III-E. This linux computer is part of an 100 Mbps ethernet network of linux nodes. One of these nodes connects to the head using two protocols. One is firewire and is used to acquire images and sound, the other is RS232 that is used to control the zoom and focus of the camera. The details about these connections are described in section III-E.

B. Small and compliant actuator

In order to have a compliant hand, we need to have compliant actuators in its joints. An actuator that complies with this requirement is a series elastic actuator (SEA) [**?**], [**?**], however, it presents problems when they are to be used in small mechanisms. Consequently we started by designing an actuator that fits our specifications.

We start by defining SEA. SEAs are comprised of an elastic element, i.e., a spring, in series with a motor (see figure 4). By measuring the deflection of the spring, one can determine the force being applied by the system. Given that the spring is the only connective element between the actuator and the output, SEAs effectively reduce the mechanical impedance of the system. This can be better explained with an example: Imagine a robotic link actuated by an SEA. Any external force applied to the link will only be resisted by the flexible spring as opposed to the high inertia projected by the gearhead reduction. Therefore, the mechanical impedance of the whole system is defined by that of the spring.

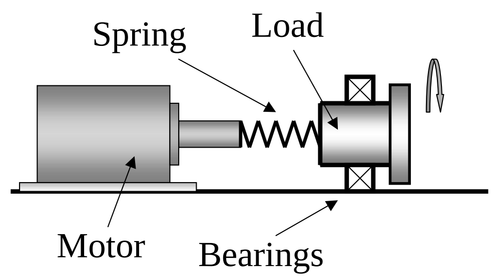

Fig. 4. Conceptual depiction of an SEA, comprising a spring in series with a motor.

Although the spring also affects the reaction speed, or bandwidth, of the system, speeds still fall within an appropriate operational range for control applications. As a physical shock absorber, the spring also makes the robotic system less susceptible to and inherently reactive to unexpected impacts.

There are both linear and rotary SEAs. The linear version requires precision ball screws to control the spring deflection. Although allowing for good mechanical transmission reduction, this constraint makes the system expensive and puts a limit on how small it can be. Conventional rotary SEAs require custom-made torsional springs, which are hard to fabricate and very stiff. This stiffness practically obviates the benefits of an elastic element. Furthermore, the torsional spring deflection is generally measured by strain-gauge sensors that are cumbersome to mount and maintain. Both of these linear and rotary SEAs present joint integration problems.

Therefore, we designed and built a different actuator that is compact, easily-mountable and cheaper to fabricate while maintaining the features of SEAs. A complete explanation of this actuator is presented in [**?**]. This actuator can be observed in figure 5.

C. Hand Design

In designing the hand we consider the following features as important:

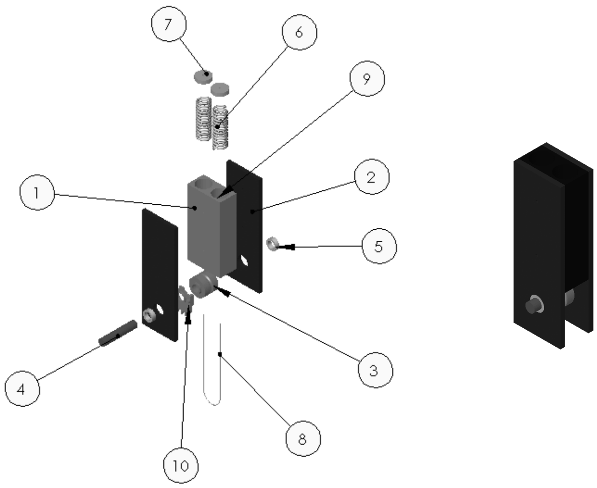

Fig. 5. The force control actuator as a whole and an exploded, annotated view.

- Flexible configuration of the fingers
- Force Sensing and Mechanical compliance
- High resolution tactile sensing.

As we describe the parts of the design, we will discuss the implementation of these features.

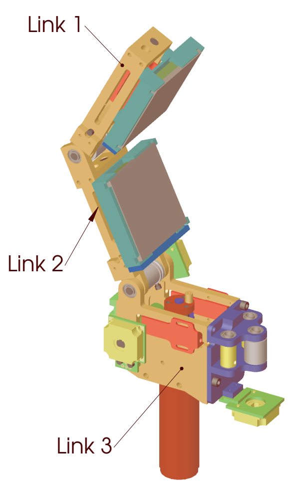

Fig. 6. CAD rendition of a finger. It comprises of three links. Link 1 and 2 have tactile sensors and their movement is coupled. Each of the three links is actuated using SEAs.

1) Finger Design: We start with the description of a finger. Each finger consists of three links as depicted in figure 6. Links 1 and 2 are coupled with a ratio of 3/4. The axes of these two links have an actuator, which is described in section III-B. This actuator has several functions: reading the torque applied to the axes, reducing the mechanical impedance of each link, and allowing the two links to decouple their movement.

The first two functions are common features of this kind of actuator and the last one is a consequence of the actuator construction.

This decoupling is useful to do grasping as described in [**?**]. For instance, we can observe in figure 7 that when link 2 contacts an object, link 1 can still keep moving to reach the object. Also link 2 is still applying force on the object.

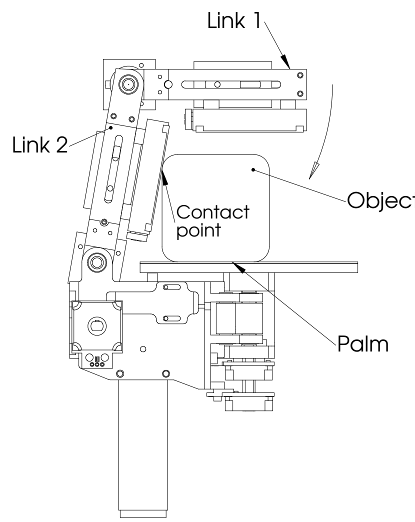

Fig. 7. Link 2 has made contact with an object and stopped moving but keeps pressing against the object. Link 1 continues moving.

In order to move links 2 and 1, there is a motor located on link 3. The torque is transmitted using cable from the motor to the the two actuators on their respective links (see figure 8). Cable is used as a transmission mechanism because unlike gears it does not have backslash problems. The different diameters of the wheels of the actuators determines the transmission ratio.

An important consideration when we are working with cables is the tension mechanism. The design of the tension mechanism in this case had to remain small so that it could fit inside link 3. We can observe it on figure 8.

In figure 8, we can also observe the presence of an idler wheel that helps to route the cable but also has a potentiometer attached to its axis to determine the absolute position of the links when they are not decoupled. When they are decoupled we need to consider the information available in the actuators.

In links 2 and 1 there are high resolution tactile sensors mounted. The details of these sensors are described in section III-C.3. On top of each sensor a rubber layer is added. This layer helps in the grabbing process given that it deforms and has good friction. The rubber chosen is the one used in tennis table paddles.

An extra feature of the finger, derived from the actuator, is the possibility of bending for pushing objects. This is clearly described in figure 9.

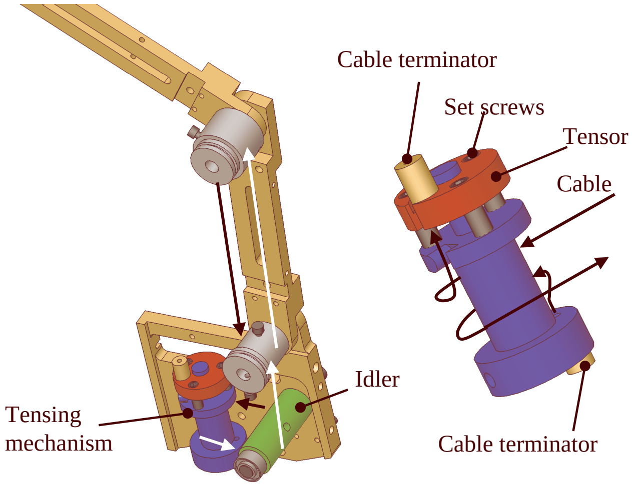

Fig. 8. On the left we can observe the cable routing in a finger. The cable comes from the tensing mechanism, goes under the idler wheel and continues to the wheels on each axis. On each of these wheels the cable is wrapped around and clamped using the screws shown on the wheels. The cable wrapped on the top wheel goes down, wraps around the lower and the idler wheel and ends on the tensing mechanism. A detail of the tensing mechanism is shown on the right of the figure. It consist of a wheel that goes connected to the motor and a lid that slides on a shaft. The cable with a terminator comes from the bottom of the wheel, continues its trajectory as described before and ends with another terminator on the lid. The lid tenses the cable by increasing the distance between itself and the wheel using the setscrews. The setscrews fit in holes that avoid rotation of the lid.

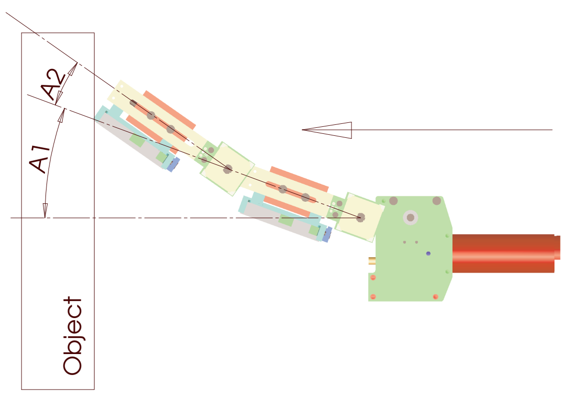

Fig. 9. When a finger pushes against an object, it passively bends and does not break thanks to the mechanical compliance of the actuators.

2) Palm and Three Fingers Design: The hand is comprised of three fingers, each like the one described above, arranged around a palm as shown in figure 10. In this configuration, finger 2 is fixed with respect to the palm but fingers 1 and 3 can move in the direction shown by the arrows. Fingers 1 and 2 can be opposed to each other as a thumb and an index finger in a human hand. Fingers 1 and 3 can also be opposed as shown in figure 11. The two degrees of freedom of the fingers around the palm allow the hand to arrange the fingers to obtain an adequate configuration for grabbing objects with a variety of shapes.

The axis of rotation of fingers 1 and 3 with respect to the palm uses a variation of the actuator described above that is used in the fingers. This provides these fingers with the advantages described earlier. The torque for each axis is provided by a DC motor which transmits movement through a cable mechanism. However, the cable tensing mechanism is a lot simpler than the one on the fingers. This is because we do not have to move coupled links, therefore, the tensing mechanism of the actuator is enough. See figure 12.

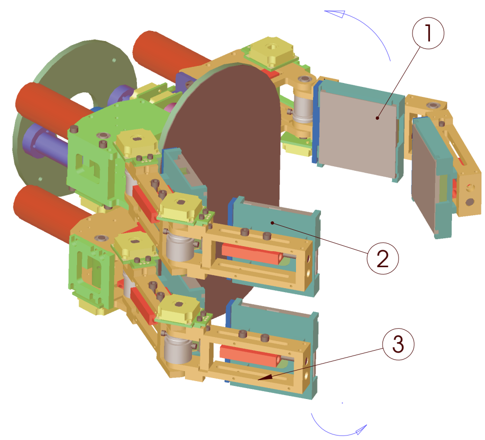

Fig. 10. This shows the arrangement of the fingers around the palm. Finger 2 is fixed to the palm while fingers 1 and 3 move up to 90° in the directions indicated by the arrows.

The palm has a high resolution tactile sensor covered with the same rubber layer as the fingers.

3) Tactile Sensor: Given that we want to use high resolution tactile sensors, we found that the best option is using a mouse pad composed of force sensing resistors (FSR). A touch pad from Interlink Electronics provides an array of FSRs whose density is 200 dots/inch and 7 bits magnitude of the force/pressure applied. The sensor reports the coordinates and the force of a point of contact. The original application of these pads is reading pen strokes from human users, therefore, the spatial resolution is high. However, when there is more than one point of contact with the pad, it reports only the average force at the center of mass of the points of contact.

The pad comes connected to a PIC microcontroller that measures the value of the resistors (FSR) and transmits the information via RS232.

The models used were VP7600 for the fingers and VP8000 for the palm.

4) Hardware Architecture: The hardware architecture for the hand consists of a DSP Motorola 56F807 that reads 7 tactile sensors, 13 potentiometers and drives 5 motors. A schematic of this architecture is shown in figure **??**.

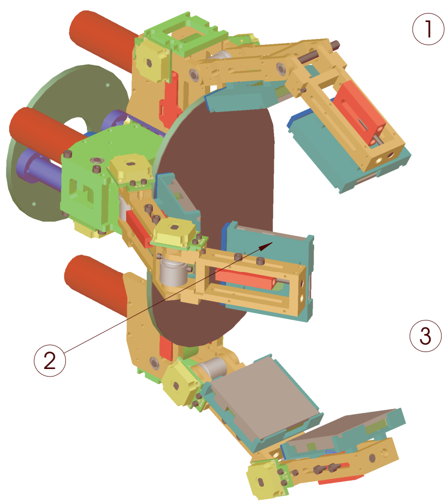

Fig. 11. The hand is shown with its fingers 1 and 3 rotated to their limit angle (90°)

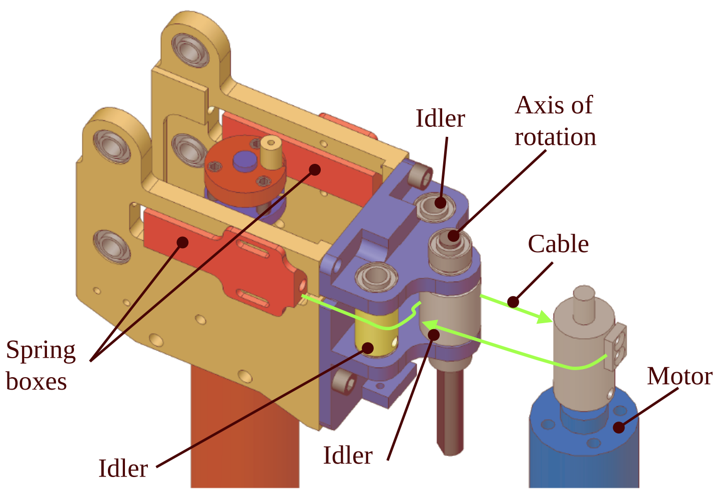

Fig. 12. We observe that the cable comes out from the spring box and turns around two idlers before getting to the motor. The idlers help to route the cable. From the motor, the cable returns to the idlers in the axle and go towards the other spring box. The spring boxes are pulled by screws placed in their back part. These screws are not shown in the figure.

Each tactile sensors sends its information to a PIC 16F877 microcontroller via RS-232. The seven microcontrollers and the DSP are arranged in a SPI network where the DSP acts as master.

The voltages from each of the potentiometers are amplified and filtered before being connected to the Analog-to-Digital (A/D) converters in the DSP.

The encoder signals from the motors are connected to a LS7266R1 chip which decodes the signal and counts the ticks. These decoders are connected to a 8 bit bus implemented by I/O lines in the DSP.

The five motors are powered by H-bridges that receive direction and PWM signals from the DSP: the direction from I/O ports and the PWM from internal generators. The connections between the DSP and the H-bridges are opto-isolated.

5) Motor Control: The low level motor control deals with force and position control of the links. A motor that controls a finger can use the force feedback from either one of the joints or position feedback from the base of the finger. For the rotation of the fingers, the feedback can come from either the position or the force feedback potentiometers. The PWM outputs were calculated using simple PD controllers updated at 10kHz. This does not exclude the possibility of combining the information from various potentiometers.

The setpoints for these controllers come from a higher level controller.

6) Hand operating: In figure 13 we can observe a sequence of picture of the hand closing on an air balloon. The hand is capable to conform with the object holding thanks to the actuators.

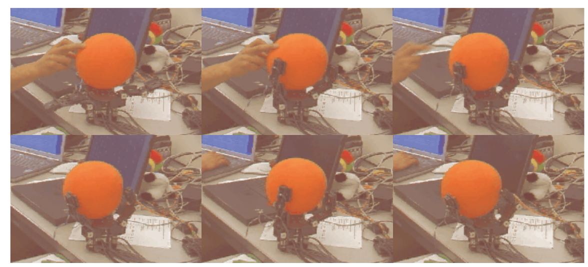

Fig. 13. Hand closing on an air balloon. The pictures are organized from left to right. On the the first to pictures (top-left) we observe the hand closing over an air balloon. When the person finger is moved, the robotic fingers and the balloon find a position of equilibrium. In the lower row, we observe that the finger in front pushes harder on the air balloon an then returns to its initial position. During that motion the other fingers maintain contact with the balloon.

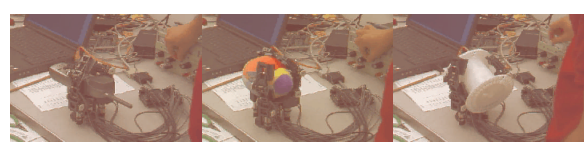

Fig. 14. Hand closing and conforming to different objects.

D. Force controlled arm

The arm used in Obrero is a copy of the arm created for the robot DOMO [**?**]. The arm has 6 DOFs: 3 in the shoulder, 1 in the elbow and 2 in the wrist. All the DOF's are force controlled using series elastic actuators.

The motor controller is similar to the one in [**?**], except for the communication module. The communication module uses an SPI physical protocol that matches the architecture described in section III-A.

E. Head: Vision and audio platform

The vision system developed is specialized for manipulation. The system was designed to take advantage of features

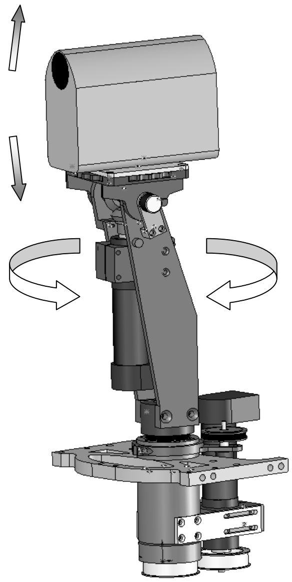

Fig. 15. Robotic Head. The head has two mechanical DOFs - pan and tilt - and two optical DOF's - zoom and focus

such as focus and zoom that are not commonly used but are very useful. Focus gives estimate of depth which is computationally less expensive. Depth information helps to position the limb. Zoom allows to get greater detail of an image. For example, we can look very closely at objects to get texture information. This is very useful when we have shadows casted.

The camera used is a Sony Camcorder model DCR-HC20 which has an optical zoom of 10 times and a resolution of 720×480 24 bit pixels. The audio system is integrated in the camcorder and provides 2 channels sampled at 44Khz. The sound and the images are transmitted to a computer using an IEEE 1394 (firewire) cable. The zoom and the focus are controlled using an RS232 port. The RS232 connects to a microcontroller PIC 16F877 that interfaces to the camcorder via LANC (Sony standard).

The camcorder is mounted on a two degree of freedom platform to get pan and tilt (see figure 15). The head is mounted in the robot torso as shown in figure 2.

The motors are controlled by a microcontroller PIC 16F877 that communicates using SPI.

The architecture of the vision and audio system is shown in figure **??**.

F. Software architecture

In order to achieve *sensitive manipulation*, we plan to use a behavior-based architecture [**?**] that let us to deal with unknown environments. Traditionally, the trajectory of the robotic manipulator is completely planned based on a model of the world (usually a CAD model). This renders the manipulator incapable of operating in a changing environment (not to mention an unknown one) unless a model of the environment is acquired in real-time.

The same situation was already faced in mobile robotics with the introduction of a behavior-based architecture that conflicted with the one based on a model of the world. However, the transition in manipulation is not straight forward.

Using a behavior-based architecture in manipulation presents other issues given the nature of the variables involved. For instance, mobile robotics uses mainly non-contact sensor (infrared, ultrasound and cameras) to determine the distance to an obstacle and act in consequence. In contrast, a manipulator needs to use mainly contact sensors (tactile and force sensors)to explore its environment. This apparently simply difference has a great consequence in the bandwidth necessary to operate the robots. Non-contact sensors give plenty of time for the robots to plan their next action even in the case of an unavoidable collision. On the contrary, contact sensors require high bandwidth. This is because when the manipulator comes in contact with an object or surface if the correct action is not taken in time either the object or the manipulator will be damaged. We can easily see this if we imagine a tactile sensor in the tip of a manipulator that intends to make contact with a table. If the acceleration of the manipulator is too high, damage will occur when contact occurs. However, the problem does not end there. Even if the manipulator makes contact with no problem, if we want to maintain the tip in contact with the table based on the information from the tactile sensor, the calculation of the kinematics of the manipulator has to be extremely precise and fast to maintain a given contact force and avoid oscillations of the tip. Some solutions to this problem involve reducing the speed of operation and padding the manipulator. These solutions render the robot unadaptable. Consequently, a behavior-based architecture is in general not an alternative for manipulation.

In order to use behavior-based architecture for manipulation, the bandwidth problem needs to be addressed. In this robot, we use passive elements to respond to the high speed components of the bandwidth. The passive elements are embedded in the actuators (SEA's) present in each degree of freedom as in Cog's arms [**?**]. This fact makes the robot an adequate platform for implementing manipulation using a behavior-based architecture.

A behavior-based architecture consists of several small modules that produce simple outputs from sensor inputs with very little processing and is not subject to a plan. These outputs are combined to obtain more complex behaviors. For example, primitive grasping reflexes, tactile feature detection, arm movements, etc. can be combined to achieve exploratory behaviors.

A tentative implementation of the behavior `lifting an unknown object` is depicted in figure 16.

In this robot, the implementation will be instantiated using tools such as L (implements a great number of light weight threads using a small amount of resources) and YARP [**?**](multiple interconnected processes running in different nodes).

1) Subsubsection Heading Here: Subsubsection text here.

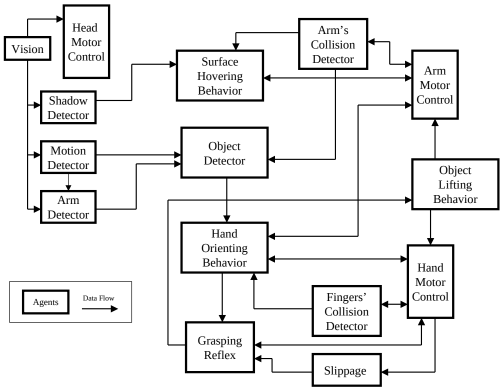

Fig. 16. Tentative implementation of the *lifting an unknown object* behavior. The *Surface Hovering* behavior moves the arm over a surface until it collides with an object. The arm's shadow is the visual cue used to maintain the arm above the surface. This behavior explores the robot's environment. The *Hand Orienting* behavior places the hand in front of an object close enough to touch the object with the fingers. The *Object Lifting* behavior grasps an object strongly enough to lift it. The combination of these three behaviors yields the *lifting an unknown object* behavior.

IV. Conclusion

In this paper we have presented the design of Obrero. Obrero is a humanoid platform built for addressing sensitive manipulation. The robot consists of a force controlled arm, a sensitive hand, and a vision and audio system.

The arm uses series elastic actuators in each of its 6 DOF. These actuators allow us to control the force applied and reduce the mechanical impedance of the arm. Reducing the mechanical impedance avoid damaging the arm and/or the object when they come in contact. This is an important feature for the design because the arm and hand will come in contact with objects often to explore the environment and grab objects, for example.

Series elastic actuators are also used to drive the hand. However, conventional versions of this type of actuator are too large and/or complicated to build as to be used in a small device like a robotic hand. Consequently, we have presented a new implementation of these actuators that reduces size, building complexity and cost.

The hand has three fingers with two links each. These links are coupled and actuated by one motor and two series elastic actuators. Two of the fingers can also rotate with respect to the palm using the same type of actuator. Each link of the fingers and the palm has high resolution tactile sensors. The purpose of these sensors is detecting features such as edges in an object or conditions such as slippage.

All these sensors in the limb allows us to treat manipulation in a different manner. For example, instead of having a model of the object, the robot, and the environment to calculate force closure for stable grasping, we can close the hand over the object and reposition the fingers until we do not detect slippage when attempting to lift it.

The vision system is intended to be a complement to the sensors in the limb as opposed to the main perceptual input. The vision system consists of a camera with control of zoom and focus. These two optical degrees of freedom are very helpful to extract information. For example, focus provides depth information while zoom helps to extract small details from an image. We try to use non-conventional visual cues from the environments such as shadows.

The low level controllers of the arm, hand and head are implemented in separate microcontrollers. All these subsystems are connected by a communication network using low overhead protocols. This configuration permits high speed communication.

In order to achieve *sensitive manipulation*, we plan to use a behavior-based architecture to deal with unknown environments given that this architecture has proven successful in mobile robots operating in unstructured and dynamic environments.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.