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.**

As a first step to achieve *sensitive manipulation* we have built Obrero, a robotic platform that addresses some of the challenges of this kind of manipulation. In this paper, we present the design, construction and evaluation of this robot.

Obrero consists of a very sensitive hand, a force controlled arm, and an active vision head. It uses non-conventional actuators, high density tactile sensors, force control and low mechanical impedance.

At mechanical level the arm and hand were designed to have low mechanical impedance and to be force controlled, which are important features when the limb come in contact with objects. These features were achieved by using series elastic actuators (SEA). A new type of SEA was conceived to operate in small spaces such as the robot's finger.

In order to increase the sensitivity of the hand, each one of the fingers and the hand palm have dense tactile sensors. These sensors allow us to detect edges, texture, motion, and other properties of the objects.

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.

At the communication level, custom hardware was designed to provided high speed communication with low overhead.

At the software level we plan to use behavior-based architecture to deal with unknown environments given that this architecture has proven successful in mobile robots operating in unstructured and dynamic environments.

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.

II. BACKGROUND LITERATURE

A. 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 [**?**].

III. ROBOT OBRERO

A. Robot Hardware Architecture

The robot Obrero is shown in figure 1 where we can observe the hand, arm, torso and head. Obrero's overall hardware architecture is presented in figure 2. In this latter figure we

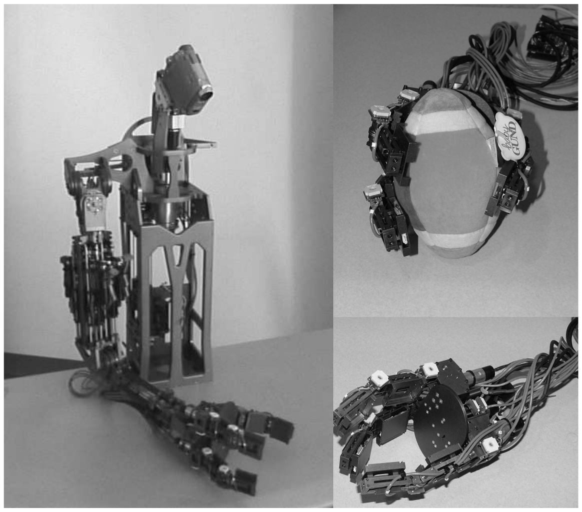

Fig. 1. Robot Obrero. The picture shows the head, arm and hand of the robot. In the upper-right corner we can observe the hand grabbing a ball.

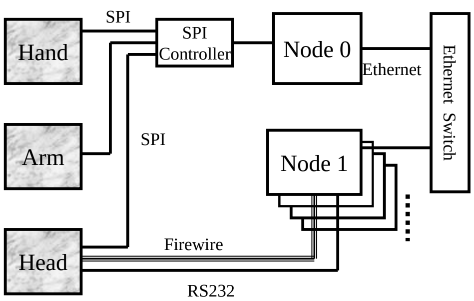

Fig. 2. Overall architecture of Obrero. The motor controllers of the Hand, Arm and Head are connected to a linux node via a SPI communication module. The head is also connected to rest of the linux network via firewire for acquiring images/sound and via RS-232 to control zoom and focus.

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 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 it is to be used in small mechanisms. Consequently we started by designing an actuator that fits our specifications.

Traditionally there were 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 3.

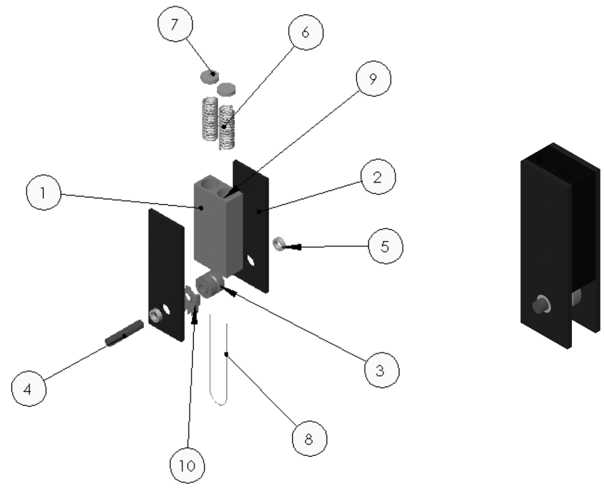

Fig. 3. The force control actuator as a whole and an exploded, annotated view.

C. Hand Design

In designing the hand we consider the following features as important: flexible configuration of the fingers, force sensing and mechanical compliance, and high resolution tactile sensing.

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

1) Finger Design: Each finger consists of three links as depicted in figure 4. 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 5 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.

In order to move links 2 and 1, there is a motor located on link 3. The torque is transmitted using cable from the

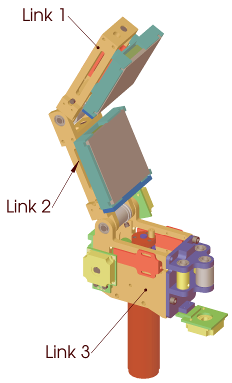

Fig. 4. CAD rendition of a finger. It comprises of three links. Link 1 and 2 have tactile sensors and their movement is coupled. Each of the three links is actuated using SEAs.

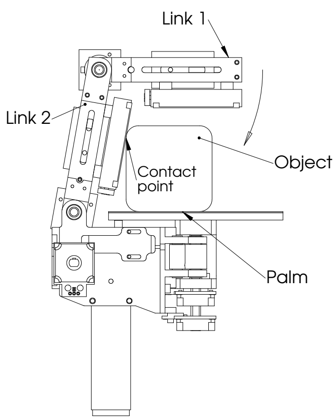

Fig. 5. Link 2 has made contact with an object and stopped moving but keeps pressing against the object. Link 1 continues moving.

motor to the the two actuators on their respective links (see figure 6). Cable is used as a transmission mechanism because unlike gears it does not have backlash 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 6.

In figure 6, 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.

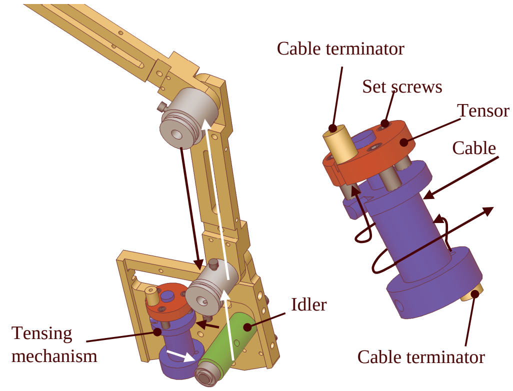

Fig. 6. On the left we can observe the cable routing in a finger. The cable comes from the tensing mechanism, goes under the idler wheel and continues to the wheels on each axis. On each of these wheels the cable is wrapped around and clamped using the screws shown on the wheels. The cable wrapped on 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.

An extra feature of the finger, derived from the actuator, is the possibility of bending for pushing objects. This is clearly described in figure 7. This features is very importatn fro dealing wiht object of low mass becasue they are dectable......

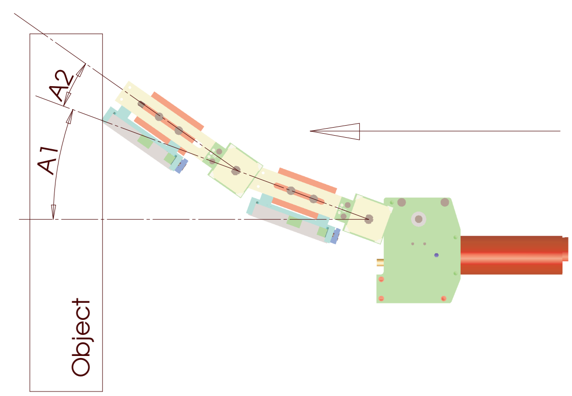

Fig. 7. When a finger pushes against an object, it passively bends and does not break thanks to the mechanical compliance of the actuators.

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 8. 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 by rotating 90°. 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 in section III-B (figure 9). 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.

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

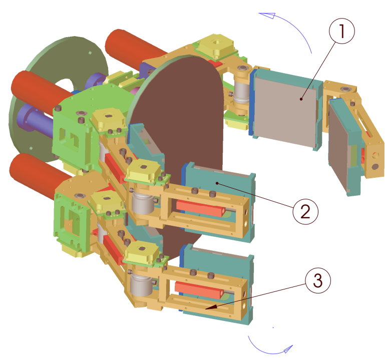

Fig. 8. This shows the arrangement of the fingers around the palm. Finger 2 is fixed to the palm while fingers 1 and 3 move up to 90° in the directions indicated by the arrows.

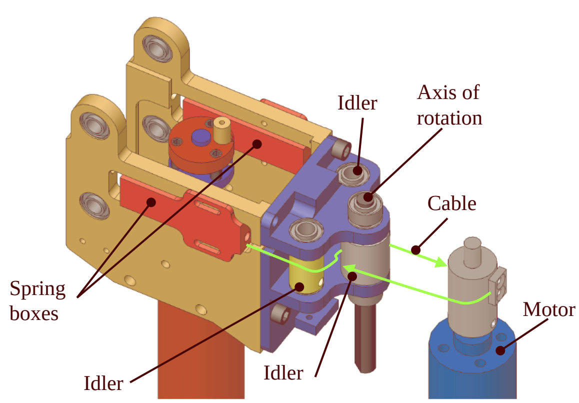

Fig. 9. We observe that the cable comes out from the spring box and turns around two idlers before getting to the motor. The idlers help to route the cable. From the motor, the cable returns to the idlers in the axle and go towards the other spring box. The spring boxes are pulled by screws placed in their back part. These screws are not shown in the figure.

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.

Each pad communicated 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.

Each tactile sensor sends its information to a PIC 16F877 microcontroller via RS-232. These seven microcontrollers reports to an eighth microcontroller via SPI and through it to 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 1kHz.

6) Hand operating: In figure 10 we can observe a sequence of picture of the hand closing on air balloon. The hand is capable to conform with the object holding thanks to the actuators.

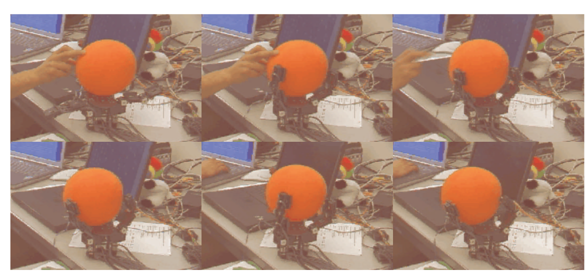

Fig. 10. Hand closing on an air balloon. The pictures are organized from left to right. On the the first to pictures (top-left) we observe the hand closing over an air balloon. When the person finger is moved, the robotic fingers and the balloon find a position of equilibrium. In the lower row, we observe 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.

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,

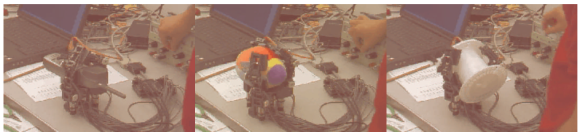

Fig. 11. Hand closing and conforming to different objects.

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

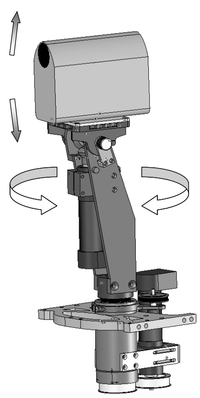

Fig. 12. Robotic Head. The head has two mechanical DOFs - pan and tilt - and two optical DOF's - zoom and focus

The vision system developed is specialized for manipulation. The system was designed to take advantage of features 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 12). The head is mounted in the robot torso as shown in figure 1.

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

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

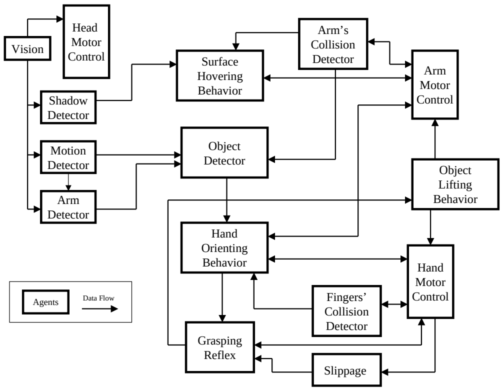

Fig. 13. Tentative implementation of the *lifting an unknown object* behavior. The *Surface Hovering* behavior moves the arm over a surface until it collides with an object. The arm's shadow is the visual cue used to maintain the arm above the surface. This behavior explores the robot's environment. 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.

behaviors.

A tentative implementation of the behavior `lifting an unknown object` is depicted in figure 13.

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).

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.