A Behavior Based Approach to Humanoid Robot Manipulation

Aaron Edsinger
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
E-mail: edsinger@csail.mit.edu

Abstract

Current approaches to humanoid robot manipulation typically rely upon detailed models of the manipulator and the object being manipulated. For robots to gain general utility in areas such as space exploration, small-parts assembly, agriculture, end even in our homes, they must be able to intelligently manipulate unknown objects in unstructured environments. A behavior based approach to manipulation can provide the tight visual and sensorimotor coupling to the world that is required during engagements in unknown and unstructured environments. To this end, we are developing an approach to humanoid robot manipulation which integrates force sensing and compliant manipulators into a behavior based control decomposition. In this paper we demonstrate work in these areas as applied to a new 29 *degree-of-freedom force controlled humanoid robot named* Domo.

1 Introduction

Today's robots are not able to manipulate objects with the skill of even a small child. For robots to gain general utility in areas such as space exploration, small-parts assembly, agriculture, end even in our homes, they must be able to intelligently manipulate unknown objects in unstructured environments. A dog can turn a bone about with two clumsy paws in order to gain a better approach for gnawing. The Osprey, or fish hawk, has a 5 DOF foot which it uses to capture prey with remarkable dexterity [26]. These animals exhibit manipulation abilities not yet attained in robots.

We contend that the manipulation efficacy exhibited by a dog is not the result of finding optimal relations between a model of its paw (which has little dexterity) and and a model of the bone (which is poorly approximated by a model). We view the dog as engaged in a tightly coupled interaction with the bone where it is modulating many different force based behaviors based on a stream of visual and tactile information. The dog's internal model of this interaction, if it is even correct to use the term 'model', is of predicted sensory consequences of the behaviors. Pushing the paw down on the bone with greater force results in less visual motion of the bone (due to increased friction between the bone and the ground). A hypothetical robotic dog doesn't need to construct a model of the bone, the paw, the ground, and the frictional forces. Instead it simply must know how to increase the paw force when the optic flow of the bone is too large.

We have seen recent successes with robots that can navigate unstructured environments. These robots, such as the Mars Sojourner, use a behavior based architecture to accommodate a dynamic and unknown environment. Manipulation hasn't yet achieved the same level of success.

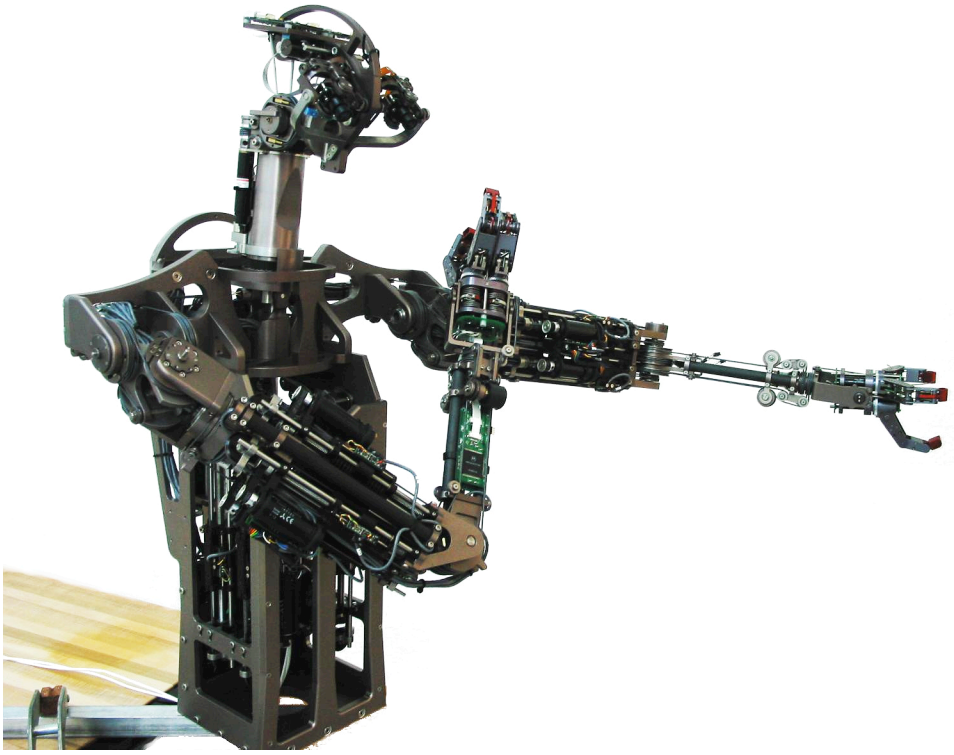


Figure 1: *The manipulation platform* Domo *being developed for this work has 29 active degrees of freedom (DOF), 58 proprioceptive sensors, and 24 tactile sensors. 22 DOF use force controlled and compliant actuators. There are two 6 DOF force controlled arms, two 4 DOF force controlled hands, a 2 DOF force controlled neck, and a 7 DOF active vision head. The real-time sensorimotor system is managed by an embedded network of five DSP controllers. The vision system includes two FireWire CCD cameras. The cognitive system runs on a small, networked cluster of PCs.*

Manipulation involves robot-environment dynamics that occur on a much shorter timescale than in navigation. Correcting a grasp to prevent dropping of an object requires tens of millisecond timescale adjustments. Navigating down a cluttered corridor requires adjustments on the timescale of hundreds of milliseconds and seconds.

One way to lessen the dominance of short-time scale dynamics is to utilize force controlled and compliant manipulators. By controlling the robot manipulator and hand in terms of joint torques instead of joint angles, we are less reliant on perceptually difficult features such as the object pose and obstacle location. Passive and active compliance in the manipulator provides local adaptation of the interaction forces between manipulator and the world without a precise world model.

In this paper we investigate a behavior based decomposition for manipulation which incorporates the modulation of a set of force-based behaviors and capitalizes on the properties of force sensing and compliant manipulators. We demonstrate this approach on a new 29 degree-of-freedom (DOF) upper-torso humanoid robot named *Domo* and describe an initial experiment in visually guided reaching and grasping.

In the next section we review the mechanical, electrical, and software systems of *Domo*. We then provide a behavior based decomposition of manipulation and review related approaches. In the following section we examine the design and control of the force sensing and compliant actuators used in the robot. Finally, we describe an initial experiment implemented on the robot which

utilizes a set of force based behaviors to reach towards and grasp a visually localized target.

2 The Robot Platform

Domo, as pictured in Figure 1, has 29 active degrees of freedom (DOF), 58 proprioceptive sensors, and 24 tactile sensors. 22 DOF use force controlled and compliant actuators. There are two 6 DOF force controlled arms, two four DOF force controlled hands, a 2 DOF force controlled neck, and a seven DOF active vision head. The real-time sensorimotor system is managed by an embedded network of five DSP controllers. The vision system includes two FireWire CCD cameras [15] and utilizes the *YARP* [12] software library for visual processing. The cognitive system runs on a small, networked cluster of PCs running the Linux operating system. We introduce *Domo*'s actuators, the SEA actuator and the FSC actuator, in the next section. See [9] for a complete description of the platform.

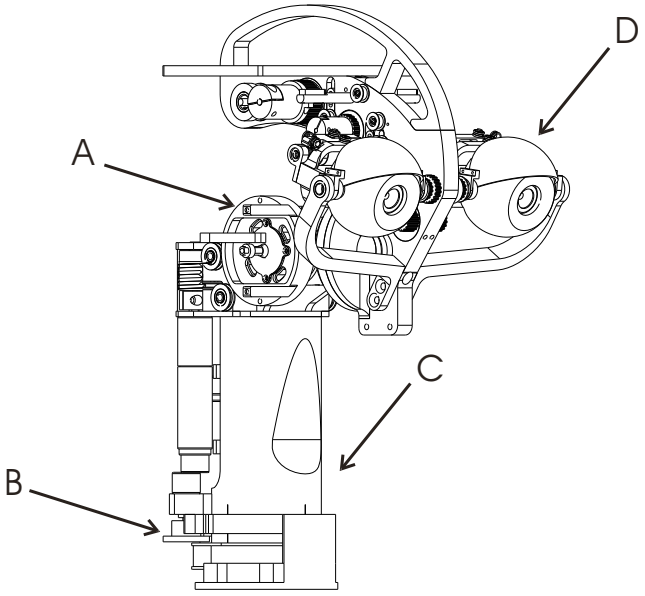


Figure 2: *Mechanical drawing of* Domo*'s active vision head. A SEA actuator driven universal joint (B) combined with head pan (C) provides a compact ball-and-socket like 3 DOF neck. The upper neck provides roll and tilt through a cable-drive differential (A). Two FireWire CCD cameras (D) share a single tilt DOF and have 2 independent pan DOF. Expressive eyelids provide a final DOF.*

2.1 Head

The design of *Domo*'s active vision head is an evolution from previous designs used for the robots Cog and Kismet [6, 3]. It is a copy of the head used in a new active vision project by Aryananda [2].

The head features 7 DOF in the upper head and a 2 DOF force controlled neck. The upper head provides roll and tilt through a compact cable-drive differential. Its two cameras share a single tilt DOF and have two independent pan DOF. The head also features one DOF expressive eyelids.

The head uses brushed DC motors with both encoder and potentiometer position feedback. The potentiometers allow for absolute measurement of position at startup, eliminating the need for startup calibration routines. The analog signal from the CCD cameras is digitized to a FireWire interface on boards mounted in the head, reducing noise issues related to running the camera signals near the head's motors. Physical stops are incorporated into all DOF to safeguard the head against potential software failures.

2.2 Arms

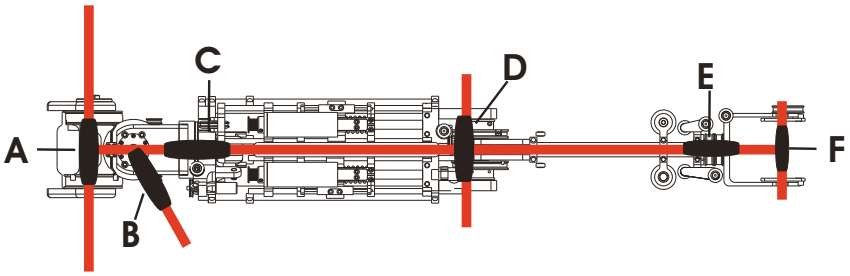


Figure 3: *The kinematic structure of* Domo*'s arms. A compact cable-drive differential at the shoulder provides pitch (A) and roll (B). These 2 DOF are driven by actuators placed in the robot torso. The bicep of the arm contains four other actuators: shoulder yaw (C), elbow pitch (D), and wrist roll (E) and pitch (F) driven by two cables routed through the elbow.*

Traditional arm designs assume that end-effector stiffness and precision are necessary qualities. A central pillar of our design approach to *Domo*'s arms is that the manipulators must be passively and actively compliant and able to directly sense and command torques at each joint.

Each arm joint is driven by an SEA actuator containing a brushless DC motor. SEA actuators and FSC actuators are covered in Section 4. As shown in Figure 3, a compact cable-drive differential at the shoulder provides pitch and roll. These 2 DOF are driven by actuators placed in the robot torso. The bicep of the arm contains another four actuators for shoulder yaw, elbow pitch, wrist roll, and wrist pitch. The drive-cables for the wrist actuators are routed through the center of the elbow joint. The cable-drive system employed through out the arm allows the actuator mass to be moved onto the torso and close to the shoulder, improving the agility of the arm.

2.3 Hands

Each of *Domo*'s hands contains four modular FSC actuator actuators acting on three fingers, as shown in Figure 4. One actuator controls the spread between two fingers. Three actuators independently control the top knuckle of each finger. The lower knuckles of the finger are passively coupled to the top knuckle. The passive compliance of the FSC actuators is advantageous. It allows the finger to better conform to an object through local, fine-grained adjustments of posture.

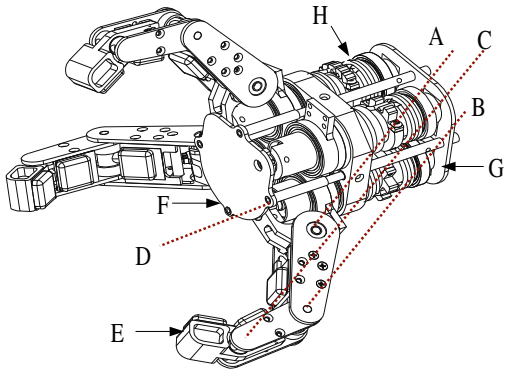


Figure 4: *Schematic drawing of the hand: Each of three fingers has three joints (A,B,C). Joint A is driven by a FSC actuator (H) through a cable drive. Joint B is passively coupled to A through a rigid cable drive. Joint C is passively linked by a compression spring to B. The spread between two of the fingers (about axis D) is driven by FSC actuator I. The interior surface of each link in a finger has a tactile sensor (E) and the palm has an array of tactile sensors (F). Electronics for motor drive, sensor conditioning, force sensing, and controller interface reside at the rear of the hand (G).*

The three fingers are mechanically identical, however two of the fingers can rotate about an axis perpendicular to the palm. These axes of rotation are mechanically coupled through gears, constraining the spread between the two fingers to be symmetric. By controlling the spread between two fingers, we can create a large variety of grasps. Force control of the spread allows for local adjustment of grasp by simply allowing the fingers to find a local force minimum.

The overall size, force capacity, and speed of the hand roughly conform to that of an human adult hand. We have modelled the kinematic structure on the Barrett Hand [28] which has demonstrated remarkable dexterity and grasping versatility.

2.4 The Sensorimotor and Software Architecture

The design of *Domo*'s sensorimotor and software architecture emphasizes robustness to common modes of failure, real-time control of time critical resources, and expansibility of computational capabilities.

These systems are organized into four broad layers: the *physical layer*, including sensors, motors, and interface electronics; the *DSP layer*, including real-time control; the *sensorimotor abstraction layer* providing an interface between the robot and the cognitive system; and the *cognitive layer*. The first two layers are physically embedded on the robot while the latter two are processes running on the Linux cluster.

The *physical layer* is made up of the electromechanical resources physically embedded in the robot. This includes: 12 brushless DC motors and amplifiers in the arms, 17 brushed DC motors and amplifiers in the hands and head, a force sensing potentiometer at each of the 22 force controlled joints, a position sensing potentiometer at each of the 29 joints, a position sensing encoder at each of the 7 joints in the upper head, and an array of 12 FSR tactile sensors in each hand.

The *DSP layer* provides real-time control over sensor signal acquisition and motor control. Each DSP node controls up to eight joints in a 1Khz control loop. It also reads up to 16 analog sensor signals at 1Khz. The arm and hand nodes provide force control loops while the head node provides position and velocity control. Higher level controllers are implemented in the *sensorimotor abstraction layer*.

The *sensorimotor abstraction layer* consists of a set of daemons running on a Linux node. It provides a substrate for the *cognitive layer* to execute on. The layer implements less time-critical motor controllers, interfaces with the CAN bus and the FireWire framegrabbers, and provides interprocess communication (IPC) infrastructure. *YARP* [12] is a robot software platform developed in our lab. It enables message based IPC distributed across multiple Linux nodes. With *YARP* we can dynamically load processes and connect them into an existing set of running processes. It allows us to communicate data at a visual frame rate of $30hz$ and a sensorimotor frame rate of $100hz$.

The *cognitive layer* is a set of distributed vision and behavior processes running on the linux cluster. The processes intercommunicate via TCP/IP underlying the *YARP* library. Each behavioral process may encapsulate several behaviors and each vision process may compute multiple percepts. The *sensorimotor abstraction layer* supports multiple process writing to the same resource, such as a joint torque command. Resource arbitration is achieved through a fixed arbitration scheme. In this way, the *cognitive layer* supports a behavior based decomposition. A behavior may subsume resource control from another behavior if it has a higher fixed priority and it is active.

3 Behavior Based Decomposition

Behavior based control architectures have been very successful in navigating mobile robots in non-laboratory environments. Architectures of this class have been used on the Mars Sojouner, the Packbot robot deployed for military operations, and in the Roomba household vacuum cleaner. The control architecture in these robots decomposes the navigation problem into a set of interacting, layered behaviors.

We maintain that today, much of robot manipulation in unstructured environments is similar to where robot navigation was 20 years ago. For example, the Stanford Cart [21] built detailed models and plans at every step during navigation. It moved one meter every ten to fifteen minutes, in lurches, and movement of natural shadows during this time would create inaccuracies in its internal model.

Similarly, much of the current work in robot manipulation uses quasistatic analysis, where detailed static models are used to compute grasp stability, for example, at each time step. This is a classic *look-think-act* decomposition where the robot senses the environment, builds a detailed model of the world, computes the optimal action to take, and then executes the action.

Real world manipulation tasks involve unstructured and dynamic environments. In this setting, explicit models and plans are unreliable. A *look-think-act* approach to manipulation, at least at the lower levels of control, renders the robot unresponsive. For example, in the time it takes a robot to build a model and compute an action, a slipping object will likely have dropped from the robot's grasp. Manipulation is characterized by a high-bandwidth coupling between the manipulator forces and the object. A behavior based decomposition provides this coupling.

3.1 Review: Behavior Based Manipulation

Some of the earliest work in behavior based manipulation was conducted by Brooks et al. with the robot Cog [6]. Cog, like our robot *Domo*, had two force controllable arms utilizing SEA actuators. It had a 7 DOF active vision head and a rudimentary force controlled gripper. The predominant work on the platform focused on active visual perception [11], multi-modal integration [1], and human imitation [27]. Williamson developed a set of rhythmic behaviors with the arms using neural oscillators [29]. However, the electromechanical robustness of the manipulators ultimately limited their utility in exploring the manipulation problem space. All of these systems were never integrated into a coherent framework.

Marjanovic [18] proposed the only truly integrative architecture for Cog. The proposed framework allows behavioral competencies to be embedded in a distributed network. The framework supports the incremental layering of new abilities and the ability to learn new behaviors by interacting with itself and the world. The learning is accomplished by autonomous generation of sensorimotor models of the robots interaction with the world. Unfortunately, the system proved perhaps too general and only simple behaviors were learned in practice. Manipulation problems were never directly addressed. However, the framework does provide an example of an integrative approach to building behavior based robots.

One of the most thorough explorations of behavior based manipulation thus far has been achieved by Grupen et al. [23], in which a hierarchical framework for humanoid robot control is proposed. Their work is tested on a humanoid platform, Dexter, which features two force-sensing Whole Arm Manipulators (WAMS) with 7 DOF each, an active vision head, and two force sensing hands with 4 DOF each.

This project decomposes the robot controller into a set of control basis behaviors, each of which is a low-dimensional sensorimotor feedback controller. These behaviors are combined by projecting the control basis of one controller onto the nullspace of the other. Novel controllers can be learned with reinforcement learning techniques. For example, a grasping policy was learned which switches between two and three fingered grasps based on the state of the grasping interaction.

They have also conducted work in incremental development of grasp controllers which do not require an a priori object model [14] and in learning haptic categories which can be used to associated visual and haptic cues with appropriate grasps [17].

The Sandini Lab has taken a developmental approach to humanoid robot manipulation, primarily with the robot Babybot [20, 22]. This robot has a single PUMA arm with coarse force control, a 16 DOF hand with only six actuators and passive compliance, and a 5 DOF active vision head. Their approach draws heavily on infant development and developmental psychology. They have developed a robot architecture based on development stages and non-model based control, fitting into our notion of a behavior based manipulation system. Natale [22] proposes an actor-critic learning scheme to function approximation of sensorimotor activity during exploratory motions. This scheme utilizes a layered set of actor-critic modules which interact in a traditional behavior based architecture. They have also investigated tightly coupled visual and motor behaviors to learn about object affordances [10]. Knowledge about the object affordances is then exploited to drive goal-directed behavior.

3.2 The Components of Dexterous Manipulation

We propose that manipulation consists of nine different components [4]. These components can be concurrent and occur multiple times during the manipulation engagement. They provide a high-level behavior based decomposition. Briefly, these components are:

1. **Deciding on actions.** A sequence of actions is determined based on the task at hand. In well characterized settings, this sequence may be determined ahead of time. Otherwise, it is generated by perceptually guided action selection mechanisms.
2. **Positioning sensors.** Sensors, such as a camera, need to be positioned to get appropriate "views" of the elements of the engagement. Positioning should occur as a result of a dynamically coupled loop between the sensor and the environment.
3. **Perception.** Perception continues throughout the engagement. Primarily the robot needs a good understanding of where things are and what objects with what properties are present before moving a manipulator in to engage.
4. **Placing body.** The pose of the robot body is adapted to allow an advantageous reach of the workspace and the ability to apply the required forces during the engagement.
5. **Grasping.** A generic dexterous hand must form a stable grip that is appropriate for any future force or transfer operations that are to be done with the object. This requires coordinating the many degrees of freedom in the hand and arm with visual and tactile perceptual streams.
6. **Force operations.** The central component of dexterous manipulation is the modulation of the interaction forces which occur between the manipulator and the object. The manipulator must apply appropriate forces and modify those forces in real time based on the response of the objects or material being manipulated.
7. **Transfer.** The manipulator transfers the object to a desired location, avoiding obstacles as necessary. This requires local knowledge of the environment.
8. **Disengaging.** The object is released from the grasp in the correct location and pose. This can be considered the inverse of grasping.
9. **Detecting failures.** The robot should detect when actions have failed. This is a perceptual problem but requires feedback into the action selection process.

We intend to implement a set of behaviors for each of these nine areas. In this paper we demonstrate initial behaviors for perception, grasping, and force operations.

4 Compliant and Force Sensitive Manipulators

Humans are very good at controlling manipulator forces, but relatively poor at controlling joint position, as demonstrated by Kawato's [13] study of arm stiffness during multi-joint movements. Joint torque in the human arm is generated by an imbalance of tension between antagonist and agonist muscles which have inherently spring-like properties. Equilibrium-point control (EPC)[19] is an influential model for arm movement which posits that the spring-like viscoelastic properties

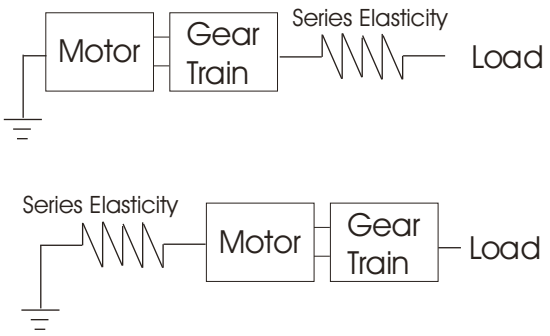


Figure 5: *Block diagram of the Series Elastic Actuator (top) and the Force Sensing Compliant Actuator (bottom). The SEA actuator places an elastic spring element between the motor output and the load. The FSC actuator places the spring element between the motor housing and the chassis ground. SEA actuators are used in* Domo*'s arms and neck. FSC actuators are used in* Domo*'s hands.*

of muscles provide mechanical stability for control. Joint posture and joint stiffness is maintained by modulating the tension of the agonist/antagonist muscle pair. EPC provides a method of arm control which does not require computing a model of the complex dynamics of the arm.

EPC is only part of the story of human arm control. However, the notion that spring and damper like qualities in the manipulator can be exploited for stable and simplified control can be leveraged for robot limb control as well.

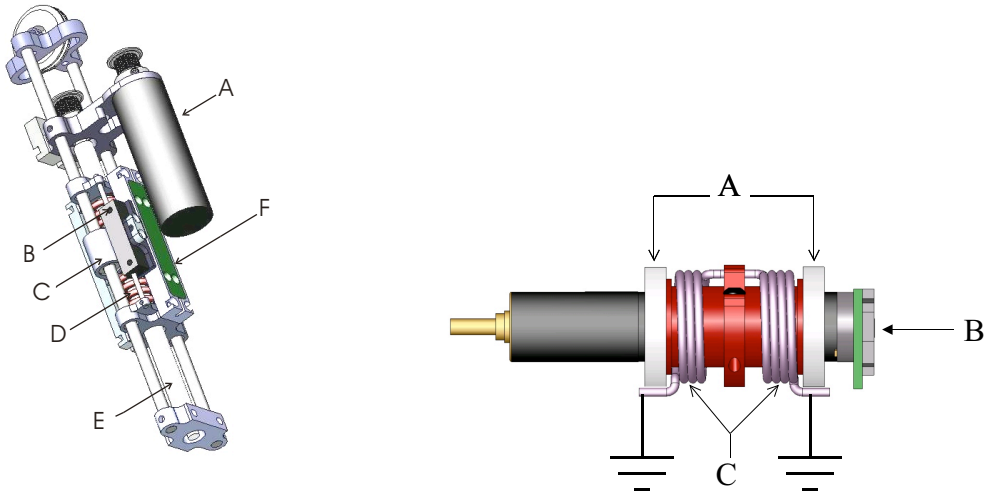
Based on previous work done on the robots Cog [6] and Spring Flamingo [25], we have built robot arms and hands specifically designed to support physical and simulated spring-damper systems. Our supposition is that, in the context of manipulation, compliant and force sensing manipulators can significantly modify the shape of the problem space to one that is simpler and more intuitive. These manipulators allow a force-based decomposition of manipulation tasks, allow the robot to safely move when the location of objects in the environment are not well known, and provide a robustness to unexpected collisions.

In this section we describe two related actuators, the Series Elastic Actuator[24] (SEA) and the Force Sensing Compliant Actuator (FSC) [8]. We have developed the FSC actuator actuator as an alternative to the SEA actuator when very compact force sensing is required. We also describe an existing method of controlling these actuators called Virtual Model Control [16].

4.1 SEA and FSC Actuators

The 20 actuators in *Domo*'s arms and hands and the two actuators in the neck utilize series elasticity to provide force sensing. We place a spring inline with the motor at each joint. We can measure the deflection of this spring with a potentiometer and know the force output by using Hooke's law ($F = -kx$ where k is the spring constant and x is the spring displacement). We apply this idea to two actuator configurations, as shown in Figure 5. The SEA actuator places the spring between the motor and the load, while the FSC actuator places the spring between the motor housing and the chassis ground. There are several advantages to these actuators:

1. The spring and potentiometer provide a mechanically simple method of force sensing.



(1) SEA actuator (2) FSC actuator

Figure 6: ***(1)*** *Model of the cable-drive SEA actuator. A brushless DC motor (A) imparts a linear motion to the inner drive carriage (C) through a precision ballscrew (E). The inner drive carriage transmits motion to the outer drive carriage (F) through two precompressed die springs (D). The deflection of the springs is measured with a linear potentiometer (B).* ***(2)*** *A simplified view of the FSC actuator. Two bearings (A) support the motor. The motor is attached to an external frame (ground) through two torsion springs (C). As the motor exerts a torque on a load, a deflection of the springs is created. This deflection is read by the torque sensing potentiometer (B).*

2. Force control stability is improved when intermittent contact with hard surfaces is made. This is an important attribute for manipulation in unknown environments.

3. Shock tolerance is improved. The use of an $N:1$ geartrain increases the reflected inertia at the motor output by N^2. This results in shock loads creating high forces on the gear teeth. The series elastic component serves as a mechanical filter of the high bandwidth forces, reducing the potential of damage to the gears.

4. The dynamic effects of the motor inertia and geartrain friction can be actively cancelled by closing a control loop around the sensed force. Consequently, we can create a highly backdrivable actuator with low-grade components.

5. The actuators exhibit passive compliance at high frequencies. Traditional force controlled actuators exhibit a large impedance at high frequencies because the motor response is insufficient to react at this timescale. In an SEA actuator, the impedance of the elastic element dominates at high frequencies.

The overall passive compliance exhibited by FSC or SEA actuators, depicted in Figure 6, is determined by the spring stiffness. If we consider that an external force applied to the actuator can only be counteracted by the spring, then we see that the mechanical impedance of the system is defined by that of the springs. The low impedance of the springs adversely affects the reaction speed, or bandwidth, of the system. For robot tasks achieved at a roughly human level bandwidth, this adverse effect is not large.

4.2 Virtual Model Control

Virtual Model Control (VMC) is an intuitive control methodology for force controlled robots. It was developed by Pratt for biped robots [16] which exhibited very naturalistic walking gaits. These robots also used the SEA actuator.

VMC represents the control problem in terms of physical metaphors that we have a good natural intuition of: springs and dampers. Virtual springs and dampers are simulated between the robot's links and between the robot and the external world. This allows force controlled movement of the manipulator with only a forward kinematic model. A dynamic model of the arm is not required.

The key idea of VMC is to add control in parallel with the natural dynamics of the arm. When we lift a milk jug into the refrigerator, we exploit the pendulum dynamics of the system to give the jug a heave. Traditional control methods override the natural dynamics of the manipulator. Instead, the manipulator follows a prescribed trajectory in joint space. A force controlled manipulator, however, can allow the natural dynamics to be exploited. Its trajectory is the composite of the natural dynamics interacting with a set of virtual springs and with the environment.

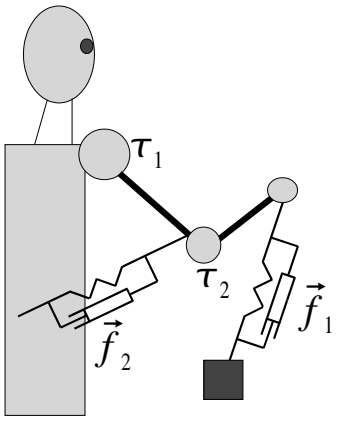


Figure 7: A simple illustration of Virtual Model Control of an arm. Virtual springs and dampers are attached between the robot body and the arm ($\vec{f_2}$) and between the end effector and the reach target ($\vec{f_1}$). With a forward kinematic model we can determine the arm Jacobian, J. These instantaneous forces can be mapped to desired joint torques: $\vec{\tau} = J^T F$

The robot Cog demonstrated exploitation of natural dynamics in a number of tasks, including sawing, hammering, and playing with a Slinky [29]. With VMC, we add layers of springs and dampers in parallel with the natural dynamics. Figure 7 illustrates an example of applying VMC to safely guide *Domo*'s arm to reach towards a target. In this simple illustration, virtual spring-dampers are used of the form $f = -k_s x + k_d \dot{x}$, where x is the spring displacement. Each spring-damper yields instantaneous forces on the arm $\vec{f_1}$ and $\vec{f_2}$. The force $\vec{f_1}$ guides the arm towards a target. The force $\vec{f_2}$ repels the elbow of the arm away from the body to avoid collisions. With a forward kinematic model we can determine the arm Jacobian, J, which relates the velocity of end-effector (or elbow) to the joint angular velocities. The end-effector force relates to the joint torque by $\vec{\tau} = J^T F$ [7].

The joint torques $\vec{\tau}$ can be commanded to the SEA actuator with a simple PID controller, simulating

the virtual springs. The stiffness of the arm can be controlled dynamically by modifying k_s, and sets of springs can be add incrementally, and in parallel, to the natural dynamics of the arm. Additionally, non-linear springs may be simulated to create specific spring behaviors.

VMC is a natural control choice for our approach to manipulation. It allows us to bias the natural dynamics of the arm through a set of independent springs acting in parallel. Consequently, we can decompose the arm controller into a set of spring behaviors, and these behaviors can augment the robot controller incrementally over time. We demonstrate this in the next section.

5 Experimental Results

We have developed an initial experiment with *Domo* involving visually guided reaching towards and grasping of an object. The experiment utilizes a single fixed camera coordinated with one arm and one hand. A rudimentary vision system designed to segment a specialized set of objects using motion cues and appearance based features provides a visual target for reaching. We use a kinematic approach to hand localization in order to predict the occurrence of the hand in the visual field. We then apply a set of force based manipulator behaviors to guide the robot hand to the target while avoiding collisions with the body. Finally, a set of grasping behaviors preshape the grasp before contact and execute the grasp based on the visual servoing error. This experiment is illustrated in Figure 8.

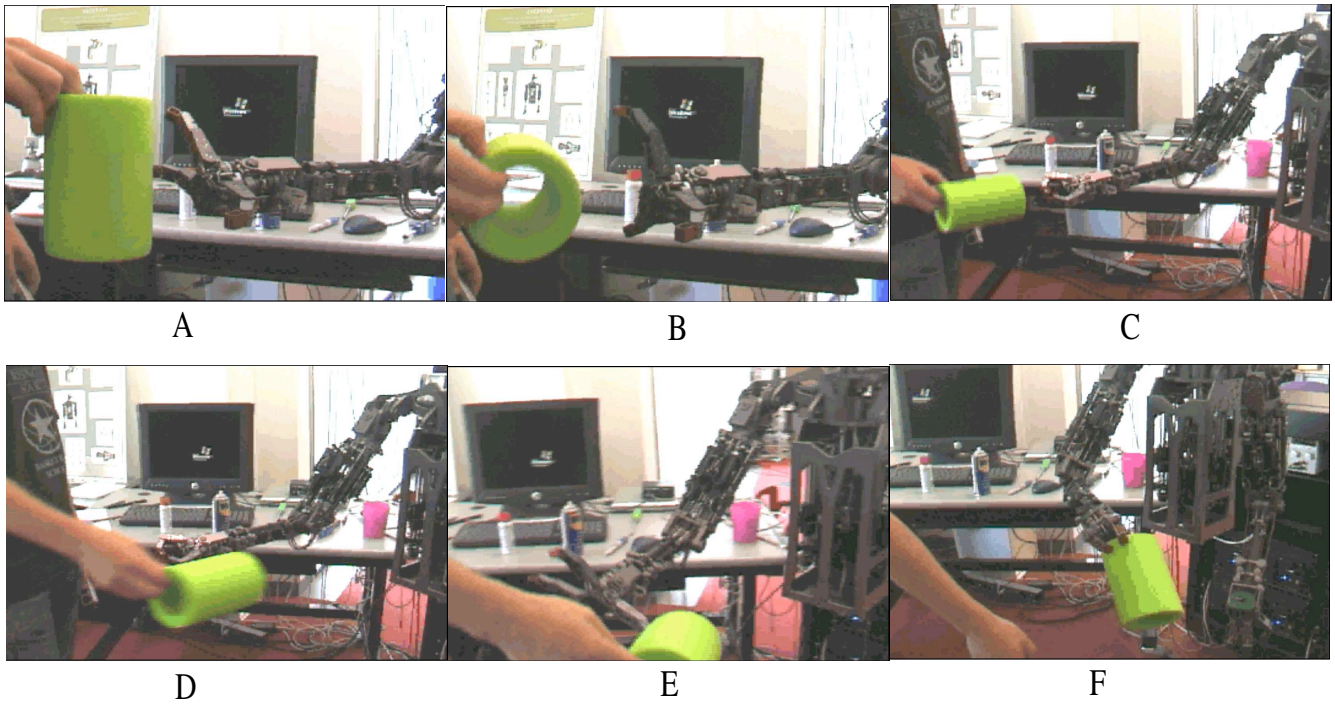


Figure 8: *A video sequence of the robot performing visually guided preshaping, reaching, and grasping.(A) The robot preshapes the grasp for a vertically oriented object. (B) The robot preshapes the grasp for a horizontally oriented object. (C,D) The start of a reaching sequence. (E)Reaching towards the target with a preshaped hand. (F) Grasp completion.*

5.1 Object Segmentation

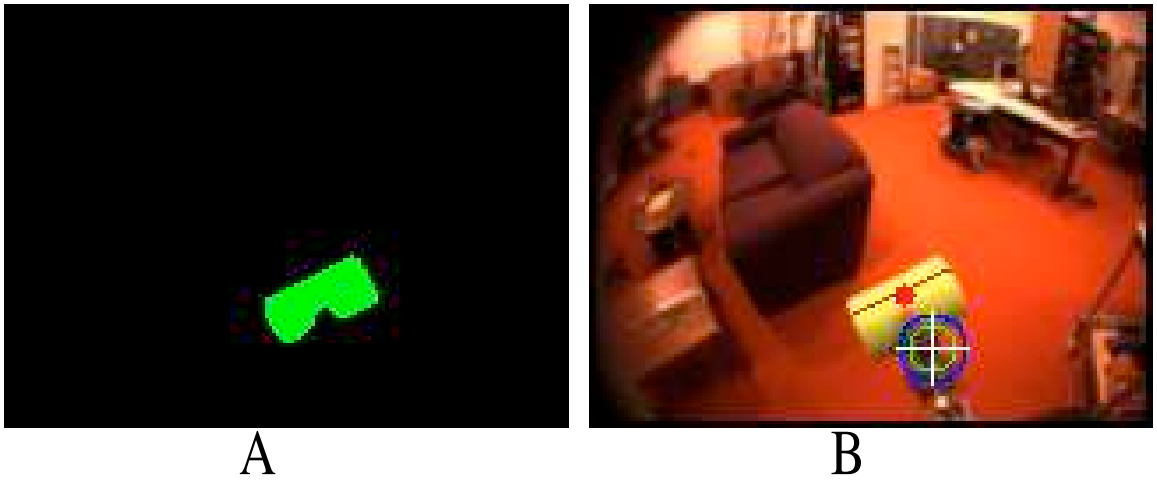


Figure 9: *(A) A simple object segmentation based on motion and color based cues. (B) The hand localization in the visual image based on a kinematic approach. The estimated hand location is represented by the cross. The target orientation axis is indicated over the target.*

We utilized a simple motion and color feature based object segmentation system to visual identify reaching targets. This is a preliminary visual system and we anticipate developing less constrained systems in the future. We employed the *YARP* vision library [12] to implement this system.

We assume a single, fixed camera and that desirable targets are of a single, saturated color (e.g., brightly colored toys). Motion in the environment is detected through background subtraction. The time averaged (1s) *RGB* value of each image pixel is subtracted from the image to separate the background from any moving objects. We then threshold filter the image for a small set of saturated colors based on precomputed color histograms. At this point we assume, optimistically, that a target, if present, has been segmented from the background. We then compute the target's primary axis of orientation based on its axis of least inertia. This orientation is used to compute an oriented bounding box for the target and also provides useful information to be used in grasp preshaping. A segmented object is illustrated in Figure 9A.

5.2 Hand Localization

We compute the occurrence of the hand in the visual image using a simple combined kinematic and lookup-table approach, which is sufficient for the purposes of this experiment. This is illustrated in Figure 9B. First, based on the rough kinematic configuration of the hand and the head, we project the manipulator endpoint on to the camera image plane. This requires only simple geometric knowledge but is error prone due to sources such as lens distortion and joint angle noise. However, it provides a good initial guess as to the hand location from which to bootstrap the localization process. In a second stage, we compensate for the localization error using a lookup-table. During a training phase, an easily identifiable marker on the hand is visually tracked. The hand is moved about its reaching workspace manually while the arm is actively backdriveable. At each sample point in the workspace, the error between the kinematic localization estimate and the visual estimate

is bound to the corresponding end-point position in the lookup-table. We constrain the error sampling to occur only in the manipulators reaching workspace. This is the area in front of the robot where manipulation activity would normally take-place. The constraint limits the size of the look-up table substantially. We also assume that the localization error is locally linear about a given point. This allows the localization error to be sparsely sampled as we can calculate the error at a given point by linear interpolation among neighboring points.

We would like to next localize the hand based on learned visual features. This requires a more sophisticated visual approach and is not addressed in this paper.

5.3 Force Based Manipulator Behaviors

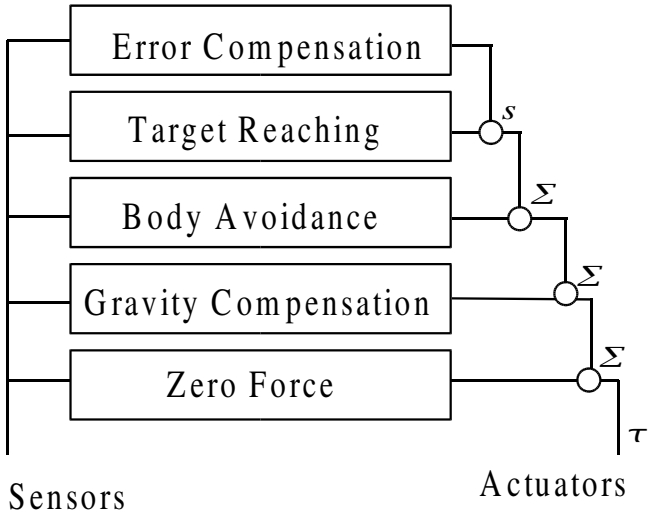


Figure 10: *A set of force based behaviors guide the robot arm to the visual target through commanded torques (τ) to each joint. The behaviors are added incrementally, augmenting the robot's ability over time. Each added behavior either adds (Σ) to the commanded actuator torque or subsumes (s) and overwrites the commanded torque . The functionality of each behavior and their interactions is described in the text.*

We implemented a set of force based behaviors to guide the robot arm to the visual target while avoiding collisions with the body. These behaviors are depicted in Figure 10. Following the traditional behavior based approach [5], these behaviors were incrementally added to the robot controller over time. At each stage the robot controller exhibited a set of coherent externally observable behaviors which expanded upon the abilities of the previous stage.

The preliminary behavior is a simple zero-force controller for the arm. By commanding a zero torque to each joint, the arm goes limp and becomes actively backdriveable. This is the default behavior for the arm in absence of any subsuming behavior.

A gravity compensation behavior is then added by summing a compensatory torque to the commanded torque of each joint. This compensatory torque, a function of the current arm posture and the arm link mass, provides a rough estimate of the gravitational loading on the arm. The gravity compensation effectively normalizes the manipulator interaction forces such that an externally applied force will have the same effect regardless of its direction. It also creates a zero-gravity mode

for the arm, allowing it to be easily moved from posture to posture about its workspace.

We then add a body avoidance behavior to the manipulator using VMC as described in Section 4.2. A set of nonlinear springs and dampers keep the arm and hand at a safe distance from the robot torso during ballistic movements. We use the following force function for the springs:

$$F = \frac{k}{2}(1 + \cos(\frac{x\pi}{c})) - d\dot{x}$$

, where k is the spring stiffness, x is the spring length, d is the damping gain, and c is the range of influence of the spring. Outside of the range, c, we set $F = 0$. The spring saturates at force $F = k$. The nonlinear aspect allows the springs' force to rapidly dissipate as the manipulator moves beyond a critical collision area. We implemented three spring behavior types: a spring between two points, a spring between a line and a point, and a spring between a plane and a point. The spring length is computed as the minimal distance between the two features. At each timestep, the virtual spring force acting on the arm is computed and translated into joint torques using the Jacobian as: $\vec{\tau} = J^T F$. Each spring is implemented as an independent behavior whose force is modulated by the nonlinear force function. We placed a virtual spring between the hand and the front body plane, a spring between the forearm and the vertical corner of the torso, and a spring between the elbow and a common collision point on the side of the torso. This set of springs allows the arm to be normally moved about in zero-gravity mode but augments this behavior when a body collision is impending.

With the preceding force behaviors, the arm can be safely moved about its workspace with low forces. We then augment the controller with a behavior to roughly reach to a target in cartesian coordinates. We should note that our visual target lacks depth information. Consequently, we assume the target exists at a fixed depth. From a forward kinematic model we know the exocentric pose of the hand. We simply attach a linear virtual spring between the hand and the target, allowing the hand to track the target in free space. Ideally, this behavior guides the hand exactly to the target. However, a steady-state error results from the rough kinematic model, non-ideal force controllers, and large stabilizing damping terms. We correct this error with an additional behavior.

The final force behavior compensates for the steady state errors by adding an additional virtual spring. Ideally we would like this compensation to be achieved by visual servoing. However, in this experiment we did not have visual hand localization information available. Instead, we added a specialized non-linear spring which moved the hand to a location just past the actual target in the direction of motion. This spring has a gaussian bell shape such that it has zero force at large distances and zero force at zero error. The peak and width of the bell is tuned to correctly compensate for the reaching error.

5.4 Grasping Behaviors

We implemented a set of simple grasping behaviors for the hand via a set of postures. Each posture is a predefined finger pose in joint space which is achieved by virtual spring control of each joint angle. The default behavior achieves a rest posture for the hand and powers off its control amplifiers.

A second behavior preshapes the grasp in anticipation of contact. The behavior is active whenever the hand and the target are both visible in the image. It subsumes postural control from the default behavior. An activation monostable prevents noise induced high frequency oscillations between

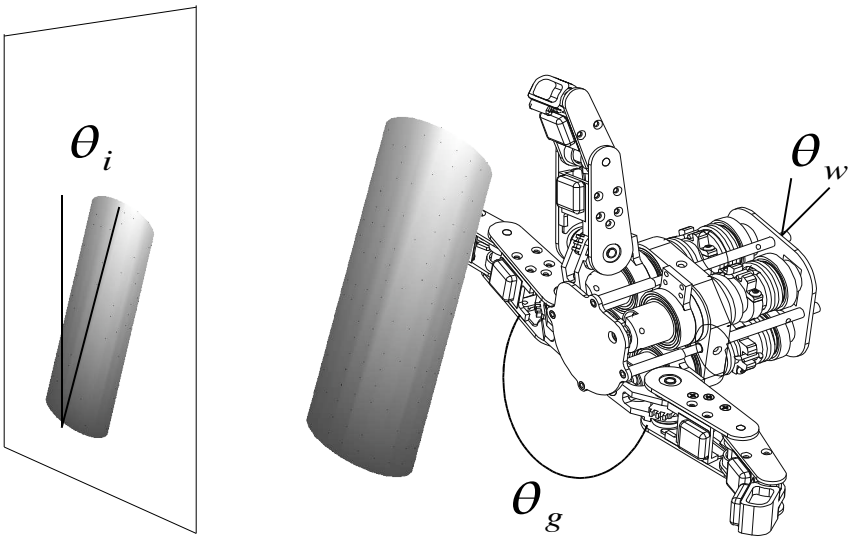


Figure 11: *The robot preshapes the hand grasp in order to advantageously precondition the manipulation interaction. The finger spread angle, θ_g, is computed based on the orientation in the image plane, θ_i and the wrist orientation, θ_w. The mapping uses θ_i and θ_w to index into a library of preshape postures.*

postures. The preshape is achieved by using the visual target orientation axis, adjusted for the wrist posture, to index into the set of postures. The preshape is controlled by the single finger spread DOF which provides the hand a limited set of postures. Figure 11 depicts this mapping.

A third grasping behavior simply achieves a closing grasp posture while maintaining the preshaped finger spread. The behavior is activated when the hand and target are within a specified pixel based distance. The tactile sensors in the hand confirm grasp of the object. If the average tactile activation over a short time window is below a threshold, the behavior is deactivated and control is relinquished back to the preshaping behavior.

6 Conclusions

We have presented a behavior based approach to humanoid manipulation as well as a description of the robot platform developed to support this approach. The robot design incorporates force sensing and compliant actuators. These allow us to directly control the torques at each joint. The actuators enable a biologically plausible approach to controlling the arm where virtual springs and dampers are simulated at the robot joints and between the manipulator and the world. This methodology for arm control supports a behavior based decomposition of manipulation tasks. We outline one such decomposition and demonstrate it in a simple manipulation experiment where the robot engages in visually guided reaching and grasping. This experiment represents a preliminary

exploration of the concepts describe in the paper. The system developed is still rather brittle to the uncertainties found in real-world environments. We anticipate expanding the perceptual abilities and the reaching and grasping behaviors available to the robot. In particular, we intend to develop visual hand localization, target depth cues, and a more general target detection scheme. We will also investigate behaviors to orient the manipulator wrist and the direction of approach used during reaching.

References

[1] A. Arsenio and P. Fitzpatrick. Exploiting cross-modal rhythm for robot perception of objects. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, December 2003.

[2] Lijin Aryananda and Jeff Weber. MERTZ: A Quest for a Robust and Scalable Active Vision Humanoid Head Robot. In *Proceedings, IEEE-RAS International Conference on Humanoid Robotics*, Santa Monica, Los Angeles, CA, USA., 2004. IEEE Press.

[3] Cynthia Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. PhD thesis, MIT, Cambridge, Ma, June 2000.

[4] Rodney Brooks, Rodric Grupen, and Robert Ambrose. Autonomous Manipulation Capabilities for Space and Surface Operations. Proposal to NASA in response to BAA-04-02, October 2004.

[5] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, April 1986.

[6] Rodney A. Brooks, Cynthia Breazeal, Matthew Marjanovic, Brian Scassellati, and Matthew M. Williamson. The Cog project: Building a humanoid robot. In C. L. Nehaniv, editor, *Computation for Metaphors, Analogy and Agents*, volume 1562 of *Springer Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1999.

[7] J. Craig. *Introduction to Robotics*. Addison Wesley, 2 edition, 1989.

[8] Aaron Edsinger-Gonzales. Design of a Compliant and Force Sensing Hand for a Humanoid Robot. In *Proceedings of the International Conference on Intelligent Manipulation and Grasping*, July 2004.

[9] Aaron Edsinger-Gonzales and Jeff Weber. Domo: A Force Sensing Humanoid Robot for Manipulation Research. In *Proceedings of the 2004 IEEE International Conference on Humanoid Robots*, Santa Monica, Los Angeles, CA, USA., 2004. IEEE Press.

[10] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning About Objects Through Action: Initial Steps Towards Artificial Cognition. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, May 2003.

[11] Paul Fitzpatrick. *From First Contact to Close Encounters: A developmentally deep perceptual system for a humanoid robot*. PhD thesis, Massachusetts Institute of Technology, 2003.

[12] Paul Fitzpatrick and Giorgio Metta. *YARP: Yet Another Robot Platform*. MIT Computer Science Artificial Intelligence Laboratory, http://sourceforge.net/projects/yarp0, 2004.

[13] H. Gomi and M. Kawato. Human arm stiffness and equilibrium-point trajectory during multi-joint muscle movement. *Biological Cybernetics*, 76:163–171, 1997.

[14] R. Grupen and C Coelho. Structure and Growth: A Model of Development for Grasping with Robot Hands. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)s*, Takamatsu, Japan, November 2000.

[15] Point Grey Research Inc. *FireFly2 IEEE-1394 CCD Camera Manual*. "http://www.ptgrey.com/products/firefly2/firefly2.pdf", 2004.

[16] P. Dilworth J. Pratt, M. Chew and G. Pratt. Virtual Model Control: An Intuitive Approach for Bipedal Locomotion. *International Journal of Robotics Research*, 20(2):129–143, 2001.

[17] Roderic A. Grupen Jefferson A. Coelho Jr., Justus H. Piater. Developing Haptic and Visual Perceptual Categories for Reaching and Grasping with a Humanoid Robot. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots*, Cambridge, MA, USA, September 2000.

[18] Matthew J. Marjanovic. *Teaching an Old Robot New Tricks: Learning Novel Tasks via Interaction with People and Things*. PhD thesis, MIT, Cambridge, Ma, 2003.

[19] J. McIntyre and E. Bizzi. Servo hypotheses for the biological control of movement. *Journal of Motor Behavior*, 25:193–203, 1993.

[20] Giorgio Metta. *Babybot: a study into sensorimotor development*. PhD thesis, LIRA-Lab, DIST, University of Genoa, 2000.

[21] Hans Moravec. The Stanford Cart and the CMU Rover. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 407–41. Springer-Verlag, 1990.

[22] Lorenzo Natale. *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping*. PhD thesis, LIRA-Lab, DIST, University of Genoa, 2004.

[23] Robert Platt, Oliver Brock, Andrew H. Fagg, Deepak R. Karuppiah, Michael T. Rosenstein, Jefferson A. Coelho Jr., Manfred Huber, Justus H. Piater, David Wheeler, and Roderic A. Grupen. A Framework For Humanoid Control and Intelligence. In *Proceedings of the 2003 IEEE International Conference on Humanoid Robots*, 2003.

[24] G. Pratt and M. Williamson. Series Elastic Actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-95)*, volume 1, pages 399–406, Pittsburg, PA, July 1995.

[25] Jerry Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. Ph.d., Massachusetts Institute of Technology, Cambridge, Massachusetts, 2000.

[26] A.M. Ramos and I.D. Walker. Raptors-Inroads to Multifingered Grasping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 467–475, 1998.

[27] Brian Scassellati. *Foundations for a Theory of Mind for a Humanoid Robot*. PhD thesis, Massachusetts Institute of Technology, 2001.

[28] William Townsend. The BarrettHand Grasper. *Industrial Robot: and International Journal*, 27(3):181–188, 2000.

[29] Matt Williamson. Neural control of rhythmic arm movements. *Neural Networks*, 11(7-8):1379–1394, 1998.