Postural primitives: Interactive Behavior for a Humanoid Robot Arm

Matthew M. Williamson*
matt@ai.mit.edu
MIT Artificial Intelligence Lab,
545 Technology Square, Room NE43-828, Cambridge, MA 02139

Abstract

This paper describes the implementation of reflex action for the arm of the humanoid robot Cog [5]. A set of biologically inspired postural primitives are used to create the arm motion. The primitives are combined in different ways to achieve reaching with grasping and withdrawal reflexes, allowing the arm to interact safely with both objects and people. This paper describes the reflexes, the biological inspiration for the control, and includes data collected from the robot.

1 Introduction

The humanoid robot Cog [5] is intended to explore its environment using its body. There are many possible ways for Cog's arms to perform this role, and this paper describes the first stage—reaching with reflexes. This allows the arm to move around safely and interact with objects and people.

Specifically, the arm reaches from a rest position to a random target (this has recently been extended to targets in visual coordinates, Marjanović at al. [18]). It will grasp—the arm stopping whenever something touches the palm, and withdraw—returning to the rest position if the top of the hand collides with anything. Also implemented is compliant motion, or a "lead" behavior, allowing the arm to be lead around to a new position, if the hand is held.

Later stages, which will lead to greater accuracy in reaching, and compensation for dynamics, are intended to be layered on top of this low-level system.

The control system used to implement this behavior has a number of different levels, both of hardware and software, and is heavily based on biological models of movement control. The arm controller design has the following biologically inspired features.

- It uses reflexes that appear in developmental stages of children (Diamond [9]).
- The arm joints have a spring-like behavior similar to that of humans.
- It controls motion using a set of postural primitives, similar to those observed in frogs and rats (Bizzi [2]).
- Conflicts and interactions between primitives are resolved using both superposition and winner-take-all, which has also been observed by Mussa-Ivaldi [23]
- It sums motion trajectories to achieve smooth motion.

The remaining sections of this paper describe in detail the biological inspiration for the arm control, the reflex network responsible for the arm behaviour, and the hardware implementation of the system. Conclusions and references to previous work are included towards the end of the paper.

2 Biological Basis

2.1 Springy Joints

It is intended for Cog's arm to be human-like, which means at least having a spring-like behaviour. There is a considerable amount of biological evidence for the spring-like behaviour of muscles (See Zajac [33] for review) and also for the spring/damper-like behaviour of individual joints (Cannon and Zahalak [6], MacKay et al. [17]). There is evidence that joints act like springs of constant stiffness, from measurements of the end-point impedance of human arms (Mussa-Ivaldi [24]). By changing the stiffness of the various muscles, the impedance (springiness) of the hand can be changed which may help to perform different tasks (Hogan [14]). In an attempt to mimic this, the actuators of Cog's arm behave as if they are springs of variable spring-rate and damping. Motion is achieved by changing the rest or equilibrium position of the springs θ_{set} as detailed in Figure 1. As shown in the figure, the torque applied to the joint τ can be written as

$$\tau = K_{joint}(\theta - \theta_{set}) + B_{joint}(\dot{\theta})$$

*Support for this research was provided JPL Contract # 959333

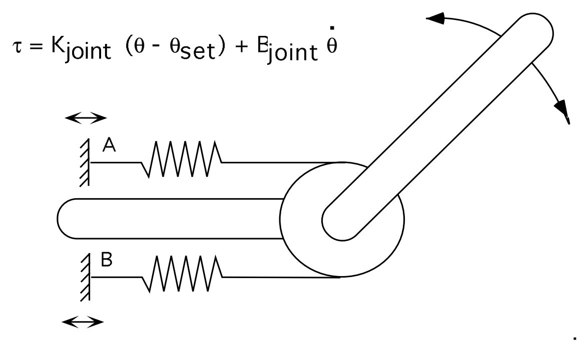

Figure 1: The spring-like actuators of Cog's arm. The diagram shows a reasonable model of the biological joint. The joint is moved by moving the end of the springs marked A and B in opposite directions . By changing the stiffnesses of the springs, the joint stiffness can be altered. The joint will have a natural equilibrium position (θ_{set}), but will generally be at a different angle (θ) due to gravitational and dynamic loads. If a damper is also included, the net torque on the joint can be given by the equation above. In Cog's arm a motor is used to produce this behavior.

This system has a sensible "natural" behaviour, if it is disturbed, or hits an obstacle, the arm simply deflects out of the way. The disturbance is dealt with by the characteristics of the system, and needs no explicit sensing or computation. Secondly since the system has a low frequency characteristic (large masses and soft springs), commands can be sent at a low rate, while still obtaining smooth arm motion. This allows more time for computation, and allows control from a system with long delays, perhaps more akin to biological systems. Thirdly, if the joint set-points are fed-forward to the arm, then the stability of the system is guaranteed.

A disadvantage of this kind of system is that it becomes more complicated to perform more traditional robotic tasks such as pure force control, (you would need to modify the set-points of the springs at a high rate) and also the low stiffness of the joints mediates against accurate position control. Humans generally achieve good position control by increasing stiffness (co-contraction) and by eliminating extra degrees of freedom (by bracing their hands while writing for example), and similar techniques would also be appropriate for Cog's arm.

2.2 Motion Primitives

Many researchers have suggested a hierarchical control scheme for movement control, based on arguments of bandwidth (there are many sensors and muscles which need a large amount of information to function correctly), delays (nerves are slow, and communicating all that information would result in long delays, with subsequent control problems), and anatomy (the loop from muscles to spinal cord to sensors suggests heavily some low level control modulated by descending signals from the brain). Experiments by Bizzi and others (Bizzi [2], Giszter [13], Mussa-Ivaldi [23], and Loeb [16]) have attempted to elucidate this hierarchy. They electrically stimulated the spinal cords of frogs and rats and measured the forces obtained at the leg, mapping out a force field in leg-motion space. This is shown in Figure 2.

Surprisingly they found that the fields were generally convergent (with a single equilibrium point) and uniform across different frogs. The idea is that if the leg is free to move, it will move to the center of the field, see Giszter [13]. Different fields therefore correspond to different postures of the leg in space. They have found only a small number of fields (4 in total), which correspond to postures either at the extremes of the workspace or to postures involved in reflex actions (such as wiping the legs against one-another).

They have found that the fields can be combined either by superposition—stimulating in two places in the spinal cord resulted in a field which was the linear vector superposition of the fields obtained when each point was stimulated, or by "winner-take-all"—stimulating in two places, and finding the resulting field coming from one of the two places (Mussa-Ivaldi [23]). They also obtained similar fields from chemical stimulation of the spinal cord, and from cutaneous simulation. The force fields remain of similar shape even after the frog is deafferented (Loeb [16]), which strongly suggests that the fields are caused by the interaction of the constant (there is no sensory feedback) spring-like properties of the muscles and the leg kinematics.

These findings lead the researchers to suggest that these fields are primitives that are combined to obtain leg motion. Mussa-Ivaldi et al. [22] have shown that fairly arbitrary force patterns, and so complex motions can be generated using a few of these force field primitives.

In Cog's arm the primitives are implemented as a set of set-points for each of the arm joints, see Figure 3. This holds the arm in a position in space. If the arm is deflected, then there is a force moving the hand back to the equilibrium position, from the action of the springy joints. A primitive is defined as a vector $P_i = (\theta_{set1}, \theta_{set2} \ldots \theta_{set6})'$ (for an arm with 6 degrees of freedom). The position of the end of the arm is generally given by the forward kinematics (see any robotics textbook such as Paul [28]). This relates the position of the end of the arm $\vec{X}$ to the joint angles θ.

$$\vec{X} = L(\theta)$$

Cog's arm is springy, so the actual joint angles are not the same as the set-point or equilibrium angles. They are related through the dynamics of the whole system

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = K_{joint}(\theta - \theta_{set}) + B_{joint}\dot{\theta}$$

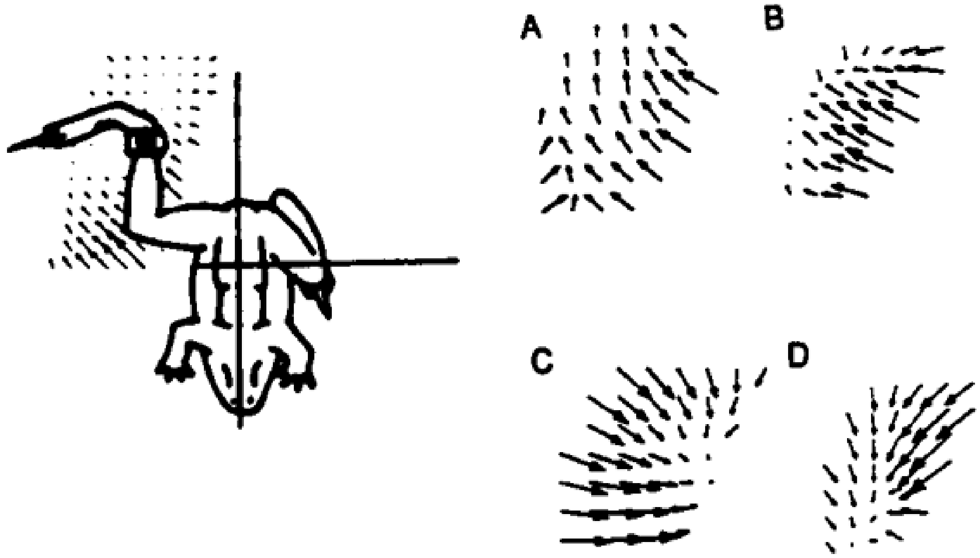

Figure 2: Convergent force fields in a frog's spinal cord. The left part of the figure shows the frog with the force transducer on its ankle. A force field is depicted under the leg. The right hand part of the figure shows the four basis fields, which are **A** leg back, **B** leg out to the side, **C** leg tucked up against the body, and **D** leg forward. Reproduced from Mussa-Ivaldi [22], with permission.

Where M, C and G have the usual meanings of inertia, coriolis and gravity terms. The arm has one equilibrium point, so in general there is another forward kinematic relation between the endpoint position $\vec{X}$ and the set-points θset.

$$\vec{X} = L(\theta) = L'(\theta_{set}) = L'(P)$$

One way to combine the primitives is by winner-take-all which means that one of the P_i has control of the arm. Alternatively linear superposition can be used:

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_n P_n$$

P also specifies a posture in space, which will be somewhere between all the P_i's. If $\sum_0^n \alpha_i = 1$ then the region of space that the arm can move in is bounded. This arrangement allows the arm to move around in a bounded region, the corners of which are the primitives, as shown in Figure 3. Unfortunately, due to the non-linearity of the forward kinematics,

$$L'(P) \neq L'(\alpha_1 P_1) + L'(\alpha_2 P_2) + \cdots + L'(\alpha_n P_n)$$

so interpolation in primitive space does not exactly match interpolation in Cartesian space. The complexity of this mapping can be greatly affected by particular choice of primitive vectors.

The force field $\vec{F}$ from a particular choice of posture is obtained by considering $\tau = J^T \vec{F}$, (see Paul [28]) and is given by the solution of

$$J(\theta)\vec{F} = K_{joint}(\theta - \theta set)$$

where $J(\theta)$ is the Jacobian from joint angles to Cartesian coordinates. This mapping produces a field with a non-linear shape.

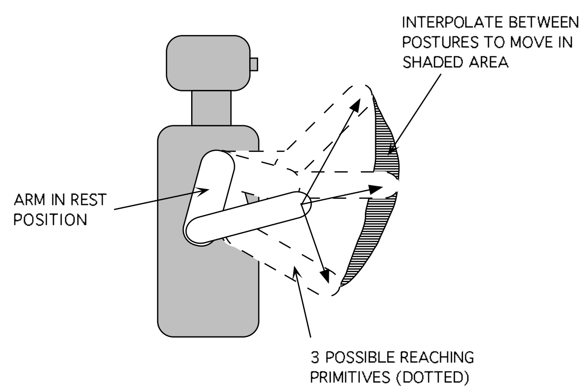

Figure 3: Primitives. Here four primitives are defined for a reaching task: a rest position, and three in front of the robot. Linear interpolation is used to reach to points in the shaded area. See also Figure 5.

The hierarchical organization has some advantages. There is a reduction in bandwidth as the commands to the arm need only set the rest positions of the springs, and do not deal with the torques directly. If a small number of primitives are used, there is also a reduction in the dimensionality of the system, with a corresponding reduction in complexity. The sacrifice made is that the arm becomes less redundant which may limit flexibility. The motion is bounded which is useful if there are known obstacles (like the body of the robot!). The primitive framework provides a very clean way of implementing the kind of withdrawal and grasping reflexes which are described in more detail in the following sections. For example, to implement the withdrawal reflex, winner-take-all is used to move the arm to the rest posture. To reach to a target, interpolation is used to move the arm to the correct position.

2.3 Smoothness and trajectory combination

The previous sections described the implementation and combination of primitives in space, whereas it is a different issue how to combine them in time. Humans when performing reaching tasks tend to move the ends of their arms in roughly straight lines, with remarkably consistent bell-shaped velocity profiles. This seems to be independent of the region of the workspace (Morasso [20], Flash and Hogan [12], Hollerbach and Flash [15]). Slow movements do tend to be more curved however (Cruse and Brüwer [7]). There has been much argument about whether these motions are planned in Cartesian space, using an optimization such as minimum jerk (Nelson [26]), or in joint space using minimum joint torque change (Uno et al [31]). Independent of what is the exact criterion, it is clear that humans move their arms smoothly.

Whenever a reflex action is initiated, the target of the arm motion changes. When humans move between a number of targets (Flash [11], Flash and Henis [12]), or make a movement that requires precision (Milner [19]), they still exhibit smooth profiles, but seem to use a number of discrete smooth movements, rather than continuous motion. This has also been observed in infants (Hofsten [32]). One conclusion is that the trajectories are summed, so that the actual motion is the sum of the old motion and the motion needed to get from the old target to the new one (Flash [12]). An alternative model is that the segmentation is an artifact of the interaction between the change in the target position and the limb characteristics (Flanagan et al. [10]).

For Cog's arms, the motion is implemented by changing the set-points of the joints using a smooth minimum jerk profile (Nelson [26]). The velocity profile for each joint is calculated from

$$\dot{\theta}_{set} = \frac{\theta_{end} - \theta_{start}}{d} \left[30(t/d)^4 - 60(t/d)^3 + 30(t/d)^2\right]$$

where t is time, and d is the duration of the move. To combine trajectories, a summation strategy is implemented. The velocities for all the active trajectories are summed and integrated to calculate the command to the arm. This allows the target to change during the motion, while still obtaining smooth motion. The new motion is from the old target to the new target, and does not require knowledge of the arm position. This is illustrated in Figure 4. The velocity trace is qualitatively similar to those presented in [11, 12, 19].

Since the interpolation is carried out in joint space, the end-point of the arm does not travel in a straight line in Cartesian space. In addition the motion is created by commanding the set-points of the arm, while the actual arm motion is dependent on gravity and dynamical loads, which can make the actual arm trajectory less smooth. This is an area where further work is needed. It is interesting that the "straightness" of infant reaches increases, as they grow older (Hofsten [32]).

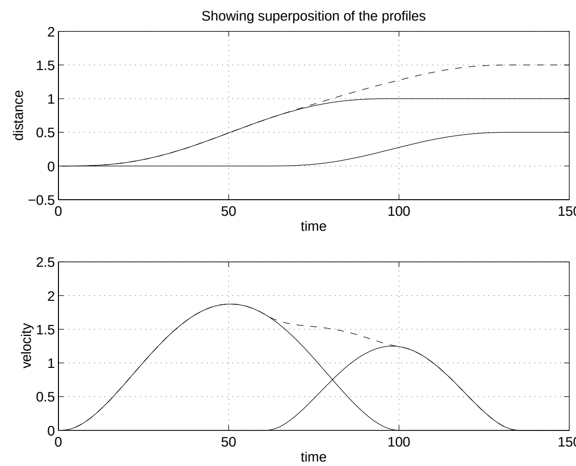

Figure 4: Implementation of summing of trajectories. The first move was from 0 to 1, then another move was added to 1.5. The top trace shows the two individual positions, and the dotted trace shows the actual motion. The lower graph shows the combination of the velocity profiles, again the dotted trace being the result of the combination.

3 Description of the reflex control

The arm is loosely based on the dimensions of a human arm, and is illustrated in Figure 5. It has 6 degrees of freedom, each powered by a DC electric motor with a series spring (Series Elastic Actuator, see Pratt and Williamson [29]). The spring is used to give good force control at the joint, and to protect the motor gearbox from shock loading. At the end of the arm is mounted a simple claw, which has touch sensors to detect when the arm is touched. These sensors then initiate the reflex responses.

The hardware setup is shown in Figure 6. Each motor is controlled using a dedicated Motorola 6811 microcontroller which runs a 1kHz control loop, creating the virtual spring behavior and managing the sensors. All the spring-like behaviour is implemented at this low level.

The 6811's communicate with a Motorola 68332 processor at about 50Hz, receiving joint set-points, control loop gains and spring parameters, and returning

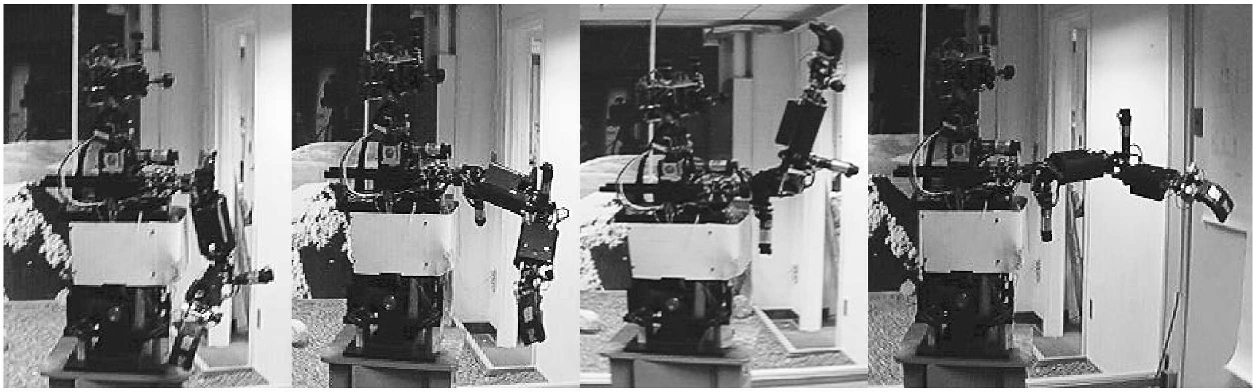

Figure 5: Picture of Cog and its arm. The four pictures also show four primitive postures, left to right, rest position, down, up and to the side. The claw at the end of the arm has 4 touch sensors, which are used by the reflexes.

torque, position, and other sensory information. The reflex network itself is implemented on another two 68332s, which communicate with each other through a shared dual-ported ram (dpram) memory interface. One of the 68332s generates targets for the reaching, and the other calculates trajectories, and deals with the sensory information. The trajectories are communicated by dpram to the communication 68332 and from there to the individual joint controllers. One final 68332 manages communication from the 68332s to a front end Macintosh computer.

The 68332s are programmed in L [4], a subset of Common LISP. This overall system architecture was chosen for Cog's brain to allow local functionality, but wide ranging communication between different parts of the brain. Indeed, this reflex system has been interfaced with the visual system getting targets for the arm motion from visual cues (Marjanović [18]).

The organization of the reflex network is shown in Figure 7. It is based on a subsumption architecture approach (Brooks [3]). The basic behavior is reaching, for which the target generator generates targets, which generate trajectories which move the arm. If the withdrawal reflex is initiated, from something touching the appropriate sensor, a new trajectory is planned to return the arm to the rest position. This new target suppresses any new targets from the target generator, and returns the arm to the rest posture. The grasp works in a similar way, adding a trajectory to return the arm to where it was first grasped, and also suppressing other targets. The lead behavior monitors the torque at each of the joints. If the behavior is initiated, it backs off the set-point of the joints if the torque goes above a threshold. Each joint is dealt with independently. It also suppresses any new targets from being processed. The trajectory generator is updated with the latest command position, so that reaching can continue as normal once the leading has finished.

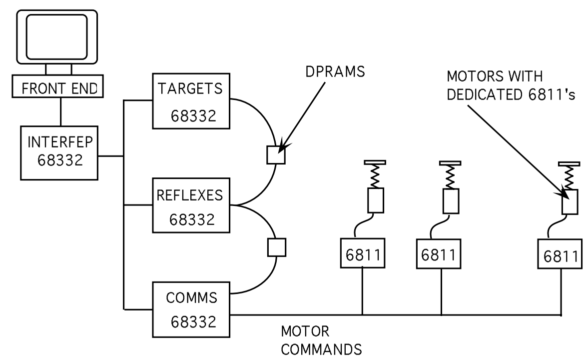

Figure 6: The arrangement of the hardware for the reaching network. Each joint has a dedicated 6811 which implements the spring-like behavior. A bank of 68332's is used to calculate targets and trajectories, hold the reflex network and communicate with the 6811's. One further 68832 manages communication between the processor bank and the Macintosh front end.

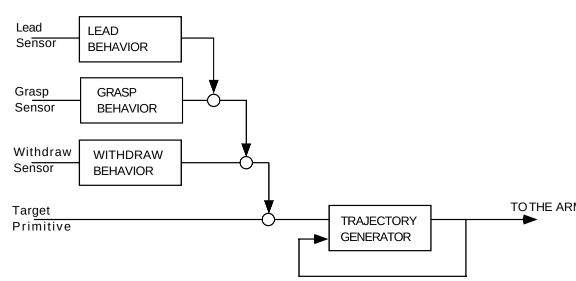

Figure 7: Figure showing the reflex network. The empty circles correspond to suppression of the lower wire.

4 Results

The arm control and reflex network worked well, and fulfilled the goal of allowing people to interact safely with the robot. The arm can be pushed around and deflected when it is stationary or moving, without causing damage.

In a complete system such as this it is difficult to find a format in which to present results and performance. This is compounded by there being no way at present to measure the position of the hand. In the results that follow, the effect of the reflex on one of the joints is described. In reality all 6 joints are moving, but the behavior of only one joint is shown for clarity.

Figure 8 shows the action of the grasp reflex. The graph shows the joint set-points, the actual joint positions and the reading on the touch sensor. The sinusoidal motion comes from the arm reaching out and returning to its rest position. The effect of the gravity loading is clear, since there is quite a large difference between the actual and the rest position of the joint. This difference also changes as the joint moves forward and up (positive on the graph) and then down again. When the grasp reflex is initiated, the arm quickly stops, and holds itself in the stopped position. The actual joint position and the set-points are different because of the gravity loading.

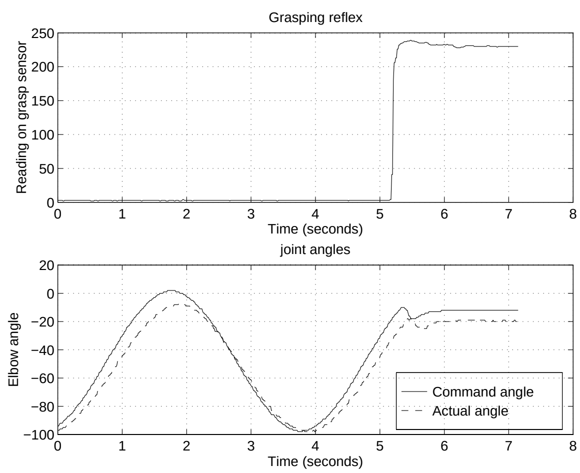

Figure 8: Grasping. The behavior of the shoulder joint is depicted in this graph. The sinusoidal motion comes from the arm reaching forward and back, until it is interrupted by the grasp reflex. The top trace shows the sensor reading, and the bottom trace show the actual (dashed) and equilibrium (solid) position of the joint. The effect of the dynamical and gravity loading is clear from the difference between the curves in the bottom trace. When the grasp is initiated, the arm quickly stops, and stays where it was originally touched.

Figure 9 shows a similar picture for the withdrawal reflex. Again the arm is moving forward and back, and when the withdraw sensor is touched, a fast trajectory back to the rest position is performed. The arm stays in the rest position for a short time, until a new target arrives from another part of the system.

Figure 10 shows the results for the leading behavior. When the hand is held, the arm is lead to a new position. This is accomplished by monitoring the torque at all of the joints, and backing off the joint equilibrium positions to keep the torque low. When leading is started, the system records the torque at all the joints. If, during the subsequent motion, the torque changes very much from that initial value, the set-point of the joints is altered accordingly. In order to compensate for the different loads that will be experienced as the arm moves through different postures, the initial torque reading is updated at a slow rate. The top graph shows the torques of the shoulder joint, as well as the slowly updating initial torque value. The bottom trace show the motion of the joint, and the changes in the set-point as the torque becomes too high. In the middle of the trace, the arm was released, and moved freely, the set-point remaining constant. The reaching behaviour was reinitiated by touching the sensor again at the end of the trace.

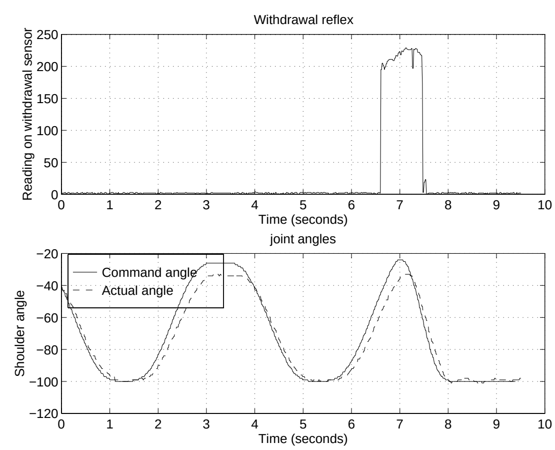

Figure 9: Withdrawal. The behavior of the shoulder joint is depicted in this graph. The sinusoidal motion comes from the arm reaching forward and back until it is interrupted by the withdrawal reflex. The top trace shows the sensor reading, and the bottom trace show the actual (dashed) and equilibrium (solid) position of the joint. When the withdraw is initiated by the high reading on the touch sensor, the arm quickly returns to the rest position.

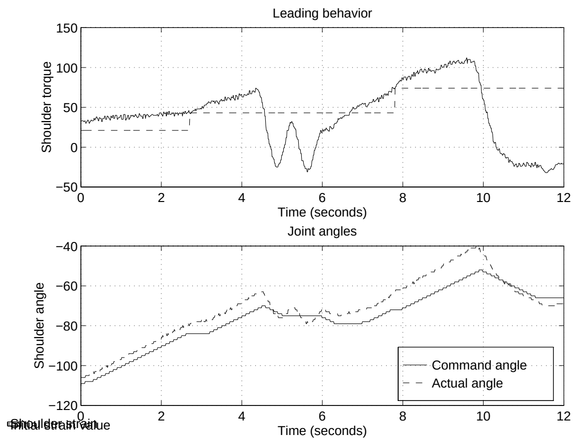

Figure 10: Leading. The graph shows the behavior of the shoulder joint as the arm is moved. The top trace shows the torque on the shoulder joint, as well as the value of the initial torque. The torque (solid line) is larger than the initial torque (dashed line) so the command angle or set-point of the joint is changed (solid line in lower graph). The initial torque reading is updated every 5 seconds to allow for changes in the torque due to postural changes. It is updated to the current torque reading (at about 3 secs, and at 8 secs). The bottom trace shows the movement of the arm (dashed) as well as the set-point (solid). In the middle of the trace (between 5 and 6 sec), the arm is released, and the leading is turned off. The set-point remains constant, and the arm can be moved without affecting the set-point (wiggles on torque and actual angle traces). At 6 seconds, the leading behavior is re-initiated, with the set-point changing as the torque changes.

5 Conclusions

This paper describes the implementation of a controller for the motion of Cog's arm. It uses sensory information to determine the motion of the whole arm. There have been fairly few approaches that have followed this path. One example is Asteroth et al. [1], who used sensory information to get a robot arm to track a moving ball. Most uses of tight sensor-motor loops in robotics have been to cope with local uncertainty during assembly, the behavior-based part commanded by a high level planner (Smithers and Malcolm [30], Morrow et al. [21], Paetsch et al. [27]). Other workers have suggested the use of hierarchical controllers and primitives for whole arm motion (Deno et al. [8], Nelson et al. [25]). The approach taken in this paper differs in that the sensory information is used to affect the overall behavior of the arm, not just modulating the planned motion of the robot end-effector.

The approach described in this paper works well, giving robust behavior to the whole arm. At the lowest level, the spring-like behavior of the joints deals cleanly with unexpected collisions, and makes the arm stable. The postural primitives which are implemented on top of that behavior make the implementation of reflexes such as withdrawal and grasping easy. The whole system is safe and certainly achieves its goal of interactability.

There are two areas which need further work, the first being compensation for dynamical loading, which will make the arm motion straighter and smoother, and the second being layering more complex motions on top of this reflex base. More complexity will need the definition of more primitives, and also the use of vision and finer touch sensing.

References

[1] Alexander Asteroth, Mark S. Fischer, Knut Möller, and Uwe Schnepf. Tracking and grasping of moving objects - a behaviour-based approach. In *Proceedings of the 5th Intl Conf IEA/AIE-92. Industrial and Engineering Applications of AI and Expert systems*, volume 1, pages 195–204, 1992.

[2] Emilio Bizzi, Ferdinando A. Mussa-Ivaldi, and Simon F. Giszter. Computations underlying the execution of movement: A biological perspective. *Science*, 253:287–291, 1991.

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

[4] Rodney A. Brooks. *The L Manual*. IS Robotics, Cambridge MA, 1995.

[5] Rodney A. Brooks and Lynn Andrea Stein. Building brains for bodies. *Autonomous Robots*, 1:7–25, 1994.

[6] S. Cannon and G. I. Zahalak. The mechanical behavior of active human skeletal muscle in small oscillations. *Journal of Biomechanics*, 15:111–121, 1982.

[7] H. Cruse and M. Brüwer. The human arm as a redundant manipulator:the control of path and joint angles. *Biological Cybernetics*, 57:137–144, 1987.

[8] D. Curtis Deno, Richard M. Murray, Kristofer S. J. Pister, and S. Shankar Sastry. Control primitives for robot systems. In *Proceedings of the 1990 IEEE Intl Conf on Robotics and Automation*, pages 1866–1871, Cincinnati, OH, 1990.

[9] Adele Diamond. Developmental time course in human infants and infant monkeys, and the neural bases of, inhibitory control in reaching. *Annals of the New York Academy of Sciences*, 608:637–676, 1990.

[10] J. Randall Flanagan, David J. Ostry, and Anatol G. Feldman. Control of trajectory modifications in

target-directed reaching. *Journal of Motor Behavior*, 25(3):140–152, 1993.

[11] Tamar Flash and Ealan Henis. Arm trajectory modifications during reaching towards visual targets. *Journal of Cognitive Neuroscience*, 3(3):220–230, 1991.

[12] Tamar Flash and Neville Hogan. The coordination of arm movements:an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, July 1985.

[13] Simon F. Giszter, Ferdinando A. Mussa-Ivaldi, and Emilio Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.

[14] Neville Hogan. The mechanics of multi-joint posture and movement control. *Biological Cybernetics*, 52:315–331, 1985.

[15] John M. Hollerbach and Tamar Flash. Dynamic interactions between limb segments during planar arm movement. *Biological Cybernetics*, 44:67–77, 1982.

[16] Eric P. Loeb, Simon F. Giszter, P. Borghesani, and Emilio Bizzi. Effects of dorsal root cut on the forces evoked by spinal microstimulation in the spinalized frog. *Somatosensory and Motor Research*, 13(2):467–491, 1993.

[17] W. A. MacKay, D. J. Crammond, H. C. Kwan, and J. T. Murphy. Measurements of human forearm posture viscoelasticity. *Journal of Biomechanics*, 19:231–238, 1986.

[18] Matthew J. Marjanović, Brian Scassallati, and Matthew M. Williamson. Self-taught visually-guided pointing for a humanoid robot. Society of Adaptive Behavior, 1996. In these proceedings.

[19] T. E. Milner. A model for the generation of movements requiring endpoint precision. *Neuroscience*, 49(2):487–496, 1992.

[20] P. Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42:223–227, 1981.

[21] J. Dan Morrow, Brad. J. Nelson, and Pradeep K. Khosla. Vision and force driven sensorimotor primitives for robotic assembly skills. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS-95)*, volume 3, pages 234–240, Pittsburg, PA, July 1995.

[22] Ferdinando A. Mussa-Ivaldi and Simon F. Giszter. Vector field approximation: a computational paradigm for motor control and learning. *Biological Cybernetics*, 67:491–500, 1992.

[23] Ferdinando A. Mussa-Ivaldi, Simon F. Giszter, and Emilio Bizzi. Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences*, 91:7534–7538, August 1994.

[24] Ferdinando A. Mussa-Ivaldi, Neville Hogan, and Emilio Bizzi. Neural, mechanical, and geometric factors subserving arm posture in humans. *Journal of Neuroscience*, 5(10):2732–2743, October 1985.

[25] R. C. Nelson, M. Jägersand, and O. Fuentes. Virtual tools: A framework for simplifying sensory-motor control in complex robotic systems. In *Proceedings of the IEEE International Workshop on Vision for Robots (IROS-95)*, pages 70–79, Pittsburg, PA, July 1995.

[26] W. L. Nelson. Physical principles for economies of skilled movements. *Biological Cybernetics*, 46:135–147, 1983.

[27] W. Paestch and G. von Wichert. Solving insertion tasks with a multifingered gripper by fumbling. In *Proceedings of the 1993 IEEE Intl Conf on Robotics and Automation*, pages 173–179, Atlanta, GA, 1993.

[28] Richard P. Paul. *Robot Manipulators: Mathematics, Programming and Control, The Computer Control of Robot Manipulators.* MIT Press, Cambridge, Massachusetts, 1981.

[29] Gill A. Pratt and Matthew 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.

[30] Tim Smithers and C. Malcolm. Programming robotic assembly in terms of task achieving behavioural modules. DAI Research Paper 417, University of Edinburgh, Edinburgh, Scotland, December 1988.

[31] Yoji Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multi-joint arm movement. *Biological Cybernetics*, 61:89–101, 1989.

[32] Claes von Hofstein. Structuring of early reaching movements: A longitudinal study. *Journal of Motor Behavior*, 23(4):280–292, 1991.

[33] Felix E. Zajac. Muscle and tendon:properties, models, scaling, and application to biomechanics and motor control. *CRC Critical Reviews of Biomedical Engineering*, 17(4):359–411, 1989.