Making Complex Articulated Agents Dance

An analysis of control methods drawn from robotics, animation, and biology

Maja J. Matarić
Computer Science Department
University of Southern California
941 West 37th Place, SAL 228, MC 0781
Los Angeles, CA 90089-0781

Victor B. Zordan
College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332-0280

Matthew M. Williamson
MIT Artificial Intelligence Lab
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139

(Received ..... ; Accepted in final form .....)

This paper will appear in *Autonomous Agents and Multi-Agent Systems*, **2(1), July 1999.**

Abstract.
We discuss the tradeoffs involved in control of complex articulated agents, and present three implemented controllers for a complex task: a physically-based humanoid torso dancing the Macarena. The three controllers are drawn from animation, biological models, and robotics, and illustrate the issues of joint-space vs. Cartesian space task specification and implementation. We evaluate the controllers along several qualitative and quantitative dimensions, considering naturalness of movement and controller flexibility. Finally, we propose a general combination approach to control, aimed at utilizing the strengths of each alternative within a general framework for addressing complex motor control of articulated agents.

Key words: articulated agent control, motor control, robotics, animation

1. Introduction

Control of humanoid agents, dynamically simulated or physical, is an extremely difficult problem due to the high dimensionality of the control space, i.e., the many degrees of freedom (DOF) and the redundancy of the system. In robotics, methods have been developed for simpler manipulators and have been gradually scaled up to more complex arms (Paul 1981, Brady, Hollerbach, Johnson, Lozano-Perez & Mason 1982) and recently to physical human-like arms (Schaal 1997, Williamson 1996). Anthropomorphic control has also found an application area in realistic, physically-based animation, where dynamic simulations of human characters, involving realistic physical models, matches the complexity of the robotics problem (Pai 1990, Hodgins, Wooten, Brogan & O'Brien 1995, van de Panne & Lamouret 1995).

In this paper, we present three controller implementations to address the tradeoffs involved in different approaches to articulated control, including joint-space control and Cartesian control, and their relevance to the different application areas, including biological models, robotics, and animation. The three controllers are implemented on a physics-based humanoid torso simulation, and applied to the task of performing a continuous sequence of smooth movements. The movement sequence chosen is the popular dance "Macarena", which provides a non-trivial, well-defined task for comparison. The particular controllers are: joint-space torque control, joint-space force-field control, and Cartesian impedance control. The paper describes each approach, and compares its performance with human data. The speed and smoothness of the resulting motions are evaluated, along with other qualitative and quantitative measures.

The rest of the paper is organized as follows. Section 2 gives the relevant background and related work in manipulator control, including biological, robotics, and animation issues. Section 3 describes Adonis, our humanoid simulation test bed. Section 4 gives a detailed specification of our task. Section 5 describes a joint-space torque controller and Section 6 describes the joint-space force-field-based controller. Section 7 contrasts those methods with a Cartesian impedance controller. Section 8 presents a detailed performance analysis and comparison of the methods along several criteria including qualitative and quantitative naturalness of appearance and controller use and flexibility. Section 9 describes our continued work toward a combination approach to articulated control, and Section 10 concludes the paper.

2. Background and Related Work

Computer animation and robotics are two primary areas of research into motion for artificial agents. This section briefly reviews each, and then introduces some biological inspiration for the types of control we will discuss.

2.1. Control in Robotics

In robotics, manipulator control has been largely, but not exclusively, addressed for point-to-point reaching. Position control of manipulators is a mature area of research offering a variety of standard techniques. A review of robotics methods can be found in Craig (1989), Paul (1981), and Brady et al. (1982). Solving the inverse kinematics (IK), or finding the relevant joint angles to obtain a desired end-point position and orientation for a given manipulator, is a difficult task, especially when the structure is redundant (Baker & Wampler II 1988). Rather than solving the inverse kinematics analytically, some techniques linearize the system kinematics about the operating point, using either the Jacobian (Salisbury 1980), or the inverse Jacobian (Whitney 1969) to achieve position control. The uses of the pseudo-inverse of the Jacobian for redundant systems has also been explored (Klein & Huang 1983).

Control methods which were originally used for force control such as hybrid position/force control (Raibert & Craig 1981), inspired work on stiffness control (Salisbury 1980) and the more general impedance control (Hogan 1985) which can be used to control the end-point position (see Section 7). Nearly all of these techniques have been augmented to include models of the robot's dynamics in order to improve the accuracy of control. The most common example is the computed torque method, where the inverse dynamics of the manipulator are solved to provide feed-forward torques during a motion (Luh, Walker & Paul 1980).

In addition, learning methods, using a variety of techniques (neural networks, fuzzy logic, adaptive control, etc.) have also been explored and continue to be applied to these problems (Atkeson 1989, Schaal & Atkeson 1994, Slotine & Li 1991, Jordan & Rumelhart 1992).

2.2. Control in Computer Animation

In computer graphics, 3D character animation has traditionally been created by hand, but recently, physical modeling has been used to automatically generate realistic motion. Current techniques for physical modeling can be classified by their level of automation; some methods minimize user-specified constraints with an automatic solver while others rely on controllers that require stronger user intervention. For example, Witkin & Kass (1988) presented a constraint-based approach with specified start and end conditions that generated motion containing characteristics such as anticipation and determination. Cohen (1992) extended this approach with higher DOF systems and more complex constraints. Ngo & Marks (1993) introduced a constraint approach to creating behaviors automatically using genetic algorithms.

Hand-tuned control of dynamic simulations has been applied successfully to more complex systems such as articulated full-body human figures. Dynamic simulation has been used to generate graphical motion by applying dynamics to physically-based models and using forward integration. Simulation ensures physically plausible motion by enforcing the laws of physics. Pai (1990) simulated walking gaits, drawing strongly from robotics work. His torso and legs use a controller based on high-level time-varying constraints. Raibert & Hodgins (1991) demonstrated rigid body dynamic simulations of legged creatures. Their hand-tuned controllers consist of state machines that cycle through rule-based constraints to perform different gaits. Hodgins et al. (1995) extended this work to human characters, suggesting a toolbox of techniques for controlling articulated human-like systems to generate athletic behaviors such as 3D running, diving, and bicycling. van de Panne & Lamouret (1995) used search techniques to find balancing controllers for human-like character locomotion, aiming at more automatic control of such simulated agents.

Other methods for generating animation automatically exist as well, including motion capture and procedural animation, but are not as relevant to the controller work presented here. For a more complete review of control in computer animation, see Badler, Barsky & Zeltzer (1991).

2.3. Control with Biological Motivation

The flexibility and efficiency of biological motion provides a desirable model for complex agent control. Our work is inspired by a specific principle derived from evidence in neuroscience. Mussa-Ivaldi & Giszter (1992), Giszter, Mussa-Ivaldi & Bizzi (1993) and related work on spinalized frogs and rats suggests the existence of force-field motor primitives that converge to single equilibrium points and produce high-level behaviors such as reaching and wiping. When a particular field is activated, the frog's leg executes a behavior and comes to rest at a position that corresponds to the equilibrium point; when two or more fields are activated, either a linear superposition of the fields is obtained, or one of the fields dominates (Mussa-Ivaldi, Giszter & Bizzi 1994). This suggests an elegant organizational principle for motor control, in which entire behaviors are coded with low-level force-fields, and may be combined into higher-level, more complex behaviors.

The idea of supplying an agent with a collection of *basis behaviors* or *primitives* representing force-fields, and combining those into a general repertoire for complex motion, is very appealing. Our previous work (Matarić 1995, Matarić 1997), inspired by the same biological results, has successfully applied the idea of basis behaviors to control of planar mobile agents/robots. This paper extends the notion to agents with more DOF. The work most similar to ours was performed by Williamson (1996) and Marjanović, Scassellati & Williamson (1996), who presented a controller for reaching with a 6-DOF robot arm, based on the same biological evidence. The system used superposition to interpolate between three reaching primitives, and one resting posture.

Another inspiration comes from psychophysical data describing what people fixate on when observing human movement. Matarić & Pomplun (1998) and Matarić & Pomplun (1997) demonstrate that when presented with videos of human finger, hand, and arm movements, observers focus on the hand, yet when asked to imitate the movements, subjects are able to reconstruct complete trajectories (even for unnatural movements involving multiple DOF) in spite of having attended to the end-point. This could suggest some form of internal models of complete behaviors or primitives for movement, which effectively encapsulate the details of low-level control. Given an appropriately designed motor controller, tasks could be specified largely by end-point positions and a few additional constraints, and the controller could generate the appropriate corresponding postures and trajectories.

3. Adonis: The Dynamic Humanoid Torso Simulation

Our chosen implementation test bed, Adonis, is a rigid-body simulation of a human torso, with static graphical legs (Figure 1), consisting of eight rigid links connected with revolute joints of one and three DOF, totaling 20 DOF. The dynamic model for Adonis was created by using methods described in Hodgins et al. (1995). Mass and moment-of-inertia information is generated from the graphical body parts and human density estimates. Equations of motion are calculated using a commercial solver, SD/Fast (Hollars, Rosenthal & Sherman 1991). The simulation acts under gravity, accepts other external forces from the environment. No collision detection, with itself or its environment, or joint limits are used in the described implementations; we have implemented these extensions in subsequent work.

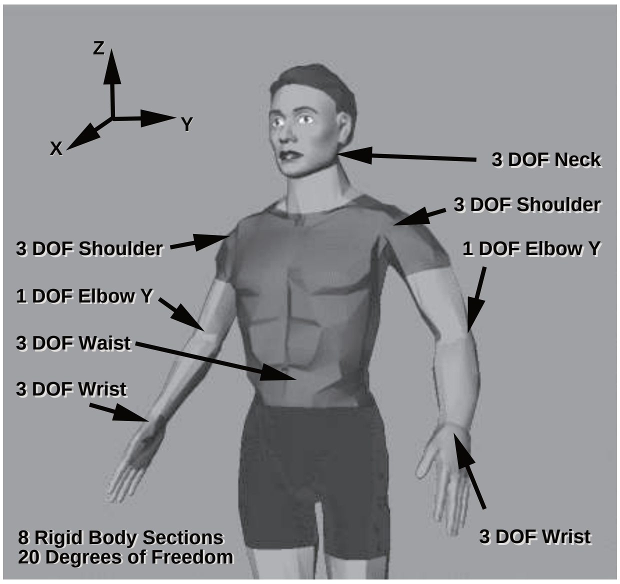

Figure 1. The Adonis dynamic simulation test bed consisting of eight rigid links connected with revolute joints of one and three DOF, totaling 20 DOF.

Adonis is particularly well-suited for testing and comparing different motor control strategies; the simulation is fairly stable and the static ground alleviates the need for explicit balance control. In addition, virtual external forces may be applied to the end-points without explicit calculation of the inverse kinematics (IK) of the arms. This, in turn, enables us to implement and evaluate experimental controllers for human-like movement more easily, while having the simulation software handle the issues of IK and dynamics. The next section introduces the task used to compare different control approaches on Adonis.

4. Task Specification

Natural, goal-driven movement relies on precise specification and coordination, and realistic constraints. As a test task should be challenging to control but familiar enough to evaluate, we chose the Macarena, a popular dance which involves a sequence of coordinated movements that constitute natural sub-tasks. We used a verbal description of the Macarena, found on the Web at *http://www.radiopro.com/macarena.html*, and aimed at teaching people the dance. Omitting the hip and whole-body sub-tasks at the end, the description is given in Table I.

Table I. The 12 sub-tasks of the Macarena.

1.	Extend right arm straight out, palm down
2.	Extend left arm straight out, palm down
3.	Rotate right hand (palm up)
4.	Rotate left hand (palm up)
5.	Touch right hand to top of your left shoulder
6.	Touch left hand to top of your right shoulder
7.	Touch right hand to the back of your head
8.	Touch left hand to the back of your head
9.	Touch right hand to the left side of your ribs
10.	Touch left hand to the right side of your ribs
11.	Move right hand to your right hip
12.	Move left hand to your left hip

This description, given as a set of sub-tasks, was used directly as the formal specification of the Macarena task. No task-level planning or sequencing was necessary because the order is provided by the dance specification. It is interesting that the individual sub-tasks are not specified in a consistent frame of reference. The first four deal with a defined posture of the whole arm, perhaps best expressed in joint angles, while the rest define the hand position, and are thus better described in an ego-centric Cartesian reference frame. As mentioned above (Section 2), people watching movement do not appear to pay active attention to the whole arm, but rather focus on the hand. However, hand position alone does not sufficiently constrain the rest of the arm, whose other joints also require specification; thus a mixture of coordinate frames is needed. This type of heterogeneous task specification is common in natural language descriptions, and control systems must satisfy each of the different goals regardless of the underlying representation. To address the issue of controller representation, we used the same Macarena specification to implement three different alternatives, described next.

5. The Joint-Space PD-Servo Approach

Joint-space controllers command torques for all actuated joints in a manipulator, and have been used successfully as low-level controllers to generate behaviors for a variety of systems (Pai 1990, Raibert & Hodgins 1991, Hodgins et al. 1995, van de Panne & Lamouret 1995). We implemented the Macarena by calculating the torques for each joint as a function of angular position and velocity errors between the feedback state and desired state, i.e., by using a hand-tuned PD-servo controller:

$$\tau = k(\theta_{desired} - \theta_{actual}) + k_d(\dot{\theta}_{desired} - \dot{\theta}_{actual}) \tag{1}$$

where k is the stiffness of the joint, k_d the damping, $\theta_{desired}, \dot{\theta}_{desired}$ are the desired angles and velocities for the joints, and $\theta_{actual}, \dot{\theta}_{actual}$ are the actual angles and velocities.

To generate the Macarena controller, the desired angles used for the feedback error are interpolated from hand-picked target postures. The postures are derived from the task specification, each corresponding to one of the 12 sub-tasks enumerated in Section 4 above. Intermediate postures between sub-tasks were used as via points to help guide the joint trajectories through difficult transitions. For example, a via point was needed for swinging the hands around the head to prevent a direct yet unacceptable path through the head. The incremental desired angles use a spline to smoothly interpolate between the postures and via points. Gains for the PD-servo are chosen by hand and remain constant through the whole Macarena.

The PD-servo approach allows direct control of each actuated joint in the system, giving the user local control of the details of each behavior. However, the controller in turn requires a complete set of desired angles at all times. Specifying that information can be tedious, especially for joints such as the neck that are less important to the behavior being generated. Interpolating between postures is a reasonable method for reducing the required amount of information. The control of actuated joints may be individually modified using their respective desired angles, thus allowing localized control over the generated motion. All desired postures are specified as a set of angles in joint-space. In the Macarena, position constraints such as "hands behind the head", can be satisfied with user-level feedback. However, precise Cartesian space constraints, like "finger on the tip of the nose", would be difficult to control with hand-tuning using joint-space errors directly. For these cases an inverse kinematics solver could be used to generate desired angles from position constraints.

6. The Joint-Space Force-Field Approach

The second implemented controller we describe is a non-linear force-field approach based on the recent work by Mussa-Ivaldi (1997), inspired by the biological data described in Section 2. In earlier work, Mussa-Ivaldi & Giszter (1992) showed that a small number of force-field primitives could be used to generate a wide range of force fields at the frog's foot. By combining the primitives using superposition, the end-point of a simulated leg could be moved to different parts of the workspace. However, the actual path taken by the leg under the influence of the field is not straight or natural looking. Subsequently, Mussa-Ivaldi (1997) showed how combinations of primitives can be used to move from one point to another in a straight line. In that work, the primitives were weighted using step and pulse functions: steps to achieve a target position, and pulses to control the trajectory of the motion.

To apply this approach to the Macarena task, stable joint-space potential fields with single static equilibrium points are combined to generate control for each sub-task. These primitives are combined with weighting functions such that step functions move the agent to its sub-task target position and pulse functions dictate desired trajectories for the arm motion, such as moving the hand to avoid the head.

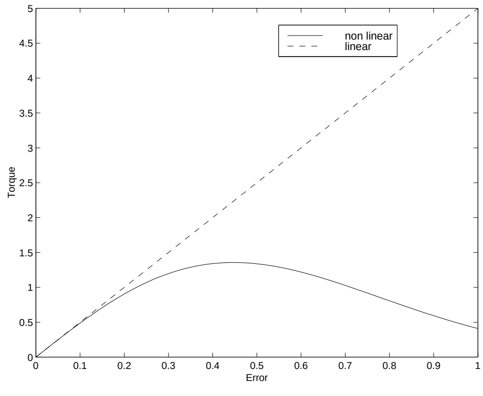

Figure 2. Graph showing the difference between the linear and non-linear joint-space controllers. The torque due to the non-linear controllers drops off at high errors.

Each primitive or force-field is specified as a torque-angle relationship at each joint of the arm:

$$\tau = \phi(t, \theta_{actual}, \dot{\theta}_{actual}) \tag{2}$$

where τ is the joint torque, and ϕ is a torque-angle relationship primitive depending on time, actual angle θ_{actual} and its derivative $\dot{\theta}_{actual}$. A primitive ϕ_i for a particular joint with stiffness k, damping k_d, and desired angle $\theta_{desired}$, is calculated as:

$$\phi_i = -k(\theta_{actual} - \theta_{desired})e^{-k(\theta_{actual}-\theta_{desired})^2/2} - k_d\dot{\theta}_{actual} \tag{3}$$

This defines a non-linear relationship, which is the derivative of a Gaussian potential centered at $\theta_{desired}$. The non-linear response of this controller is similar to a linear PD-servo for small errors $(\theta_{actual} - \theta_{desired})$. However, with large errors, the torque calculated by the primitive drops off exponentially, as shown in Figure 2. Mussa-Ivaldi & Giszter (1992) suggest that this behavior is consistent with biological muscle, and that the non-linearity of the controller increases the richness of behavior that can be produced.

We specified each sub-task of the Macarena with two such non-linear primitives combined to create the whole motion. The two primitives perform different tasks: the static position, defined by a force-field ϕ_i weighted by a step function $\omega_i(t)$, and the path between sub-tasks, manipulated using another force-field ψ_i, itself weighted by a pulse function $v_i(t)$:

$$\tau = \omega_i(t)\phi_i(t, \theta_{actual}, \dot{\theta}_{actual}) + v_i(t)\psi_i(t, \theta_{actual}, \dot{\theta}_{actual}) \tag{4}$$

The step function is defined by:

$$\omega_i(t) = \begin{cases} t - (1/2\pi)sin(2\pi t) & \text{if } 0 < t < 1 \\ 0 & \text{if } t \leq 0 \\ 1 & \text{if } t \geq 1 \end{cases} \tag{5}$$

which yields a smooth transition in the control corresponding to movement toward a particular final posture defined by $\theta_{desired}$. The pulse function is defined by:

$$v_i(t) = \begin{cases} 1 - cos(2\pi t) & \text{if } 0 < t < 1 \\ 0 & \text{if } t \leq 0 \text{ or } t \geq 1 \end{cases} \tag{6}$$

which creates a smooth adjustment in the trajectory allowing separate control of the path taken in the movement.

Our implementation differs from Mussa-Ivaldi (1997) in a number of ways. Mussa-Ivaldi uses a set of arbitrarily chosen primitives, and solves a least squares optimization problem to determine the sizes of the steps and pulses. Rather than select arbitrary primitives, we chose ours to correspond to the positions of the arm at each sub-task, thus simplifying the weighting. This is a pragmatic decision; it is unclear how well the optimization method scales from the 2 DOF system implemented in the Mussa-Ivaldi paper, to the full 20 DOF Adonis simulation. Finally, in the Mussa-Ivaldi work the primitives are defined as a Gaussian potential in the full joint-space, coupling the joints, while in our implementation they are treated independently.[1]

This force-field-based joint-space controller (heretofore referred to as the torque-field controller) is similar to the PD-servo joint-space controller described in the previous section, in that they both rely on torque-angle relationships at the joints to determine the arm motion. The main difference

[1] Mussa-Ivaldi defines ϕ_i as

$$\phi_i = -k(\theta_{actual} - \theta_{desired})e^{(-k\sum_{joints}(\theta_{actual}-\theta_{desired})^2)/2} - b\dot{\theta}_{actual}$$

which couples the joints through the exponential term.

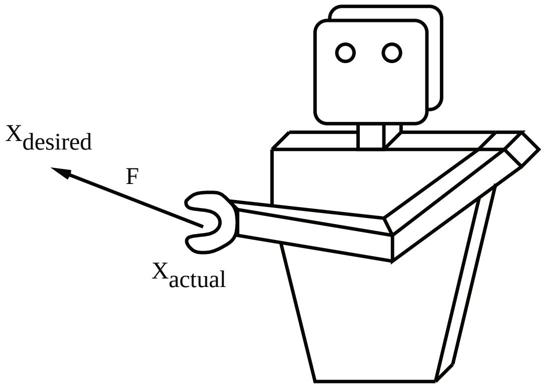

Figure 3. Impedance Control: The virtual force F is computed by attaching a virtual spring and damper from the hand position x to the desired position x_e. The torques at the joints are then calculated to produce this desired force at the end of the arm, and thus move it to the desired position.

is that the torque-field approach uses non-linear controllers at the joints, as opposed to the linear PD-servos. This non-linearity allows the controller to simply switch set-points for a new task, rather than interpolate as in the linear case, and to use pulse functions to manipulate the trajectory, rather than define explicit via points.

7. The Cartesian Impedance Control Approach

In contrast to the first two, our third implemented controller acts in the Cartesian frame of reference, which allows for a more intuitive interface for the user, as the Cartesian position of the hand is easier to visualize than the angles of all the joints. The approach is based on the principle of impedance control, introduced by Hogan (1985), has been applied to robot manipulation. The general principle is to modulate the mechanical impedance of the end-point of an arm by altering the torques at the arm's joints. Mechanical impedance for an object is defined as the relationship between an imposed disturbance and a generated force. For example, a compressed spring exerts a force proportional to the displacement. The impedance of such a system is constant and equal to the stiffness of the spring. For a more complicated mechanism like a robot arm, the mechanical impedance is determined by the control at the joint level. For example, a mechanical arm can be made to appear as if a virtual spring and damper are connected to some equilibrium point; moving the point will drag the arm around, and the arm will automatically return to its equilibrium position if disturbed. Arranging the control of the arm in this way has advantages in terms of stability, especially when interacting with different environments (Colgate & Hogan 1988).

Our impedance controller calculates the force F from the virtual spring and damper, as illustrated in Figure 3, given by:

$$F = K(x_{desired} - x_{actual}) - B(\dot{x}_{desired} - \dot{x}_{actual}) \tag{7}$$

where x_{actual} is the 6-D vector defining the position and orientation of the end-point (hand) in space, $\dot{x}_{actual}$ is a vector of velocities, and $x_{desired}$ and $\dot{x}_{desired}$ are 6-D vectors of desired positions/orientations and velocities. K and B are stiffness and damping matrices. This desired force is implemented by applying torques τ at the joints, which are calculated using the Jacobian $J(\theta_{actual})$, using the following simple relation (Craig 1989):

$$\tau = J(\theta_{actual})^T F \tag{8}$$

Applying this equation results in stable control of the position and orientation of the hand over the workspace of the arms. However, it does not constrain the final orientation of the whole arm, or prevent the arm from violating joint limits or moving through the body. To further constrain the arm, a second impedance controller was added to control the elbow motion. This allows the positions of the elbow and the hand to be moved, which is an intuitively sensible method of constraining the arm motion. Experiments showed that the best way to control the elbow was to specify a desired orientation for the upper arm, rather than specifying the elbow position.[2] The control is calculated in a similar manner to Equation 8, although the Jacobian is defined for the transformations between the elbow and 3D shoulder joint, and the force F is only due to desired rotations. Other terms added to the impedance control include compensation for the effect of gravity on the links of the arms ($g(\theta_{actual})$), and some extra damping at the shoulder joint ($b_{shoulder}$), making the final torque applied to the joints:

$$\tau = J_{hand}^T F_{hand} + J_{elbow}^T F_{elbow} + g(\theta_{actual}) + b_{shoulder} \tag{9}$$

To perform each sub-task of the Macarena, we specify the desired position and orientation of the hand, and the desired orientation of the upper arm. The control scheme then calculates the torques at the joints in order to move the arm to that position, and maintain it there. Low-level PD-servos, as described previously, control the waist and neck. To move between sub-tasks, a linear interpolation scheme is used to gradually shift the desired positions. As with the PD-servo controller, extra via points are used to avoid collisions with the head.

The method has several advantages over position control techniques using inverse kinematics (Baker & Wampler II 1988). It is computationally simple, requiring only the forward kinematics and the Jacobian (Whitney 1982), and it is stable both when moving freely, and during contact with surfaces (Hogan 1985). In addition, the general formulation of impedance control provides a simple merging mechanism for different control strategies (Beccari & Stramigioli 1998).

The main difficulty encountered when implementing this scheme was finding a compact and intuitive way to specify the orientations of the elbow and hand. The orientation of the hand was specified using a single angle relative to the lower arm, while the orientation of the upper arm was specified by aligning the x-axis of the segment with a desired vector. In addition, the scheme produces straight-line motions of the hand which are not always the most natural. For example, when moving the hand from straight out (sub-task 3) to touching the shoulder (sub-task 5), the most natural motion is for the hand to come up and over, rather than moving directly in a straight line. A curved solution is possible with this controller, but would require a more detailed specification of the desired trajectory.

As an alternative to impedance control, the simulation system allows arbitrary forces to be applied to the end-point of the arm. Thus a force could be calculated as in Equation 7, and directly applied to the hand. A variant of this approach was experimented with, applying the following force:

$$F = c(v_{actual} - v_{desired})|x_{actual} - x_{desired}| \tag{10}$$

where $v_{desired}$ is the desired velocity, v_{actual} is the actual velocity, x defined as above, and c is a gain constant. For carefully chosen values of c, this controller has the effect of moving the hand to the desired Cartesian position $x_{desired}$. Although simpler to implement than the impedance controller, this controller has a number of disadvantages. Since the force is only applied at the hand, high damping has to be used to constrain the rest of the arm, which results in unnatural motion. The

[2] This is due to the fact that under impedance control, the arm moved under the influence of the applied virtual springs and dampers at the hand and elbow. The effect of two forces on the arm can be unintuitive for arbitrary positioning of the set-points. Specifying the orientation of the upper arm, as well as the position and orientation of the hand, makes the system much more predictable and easy to operate.

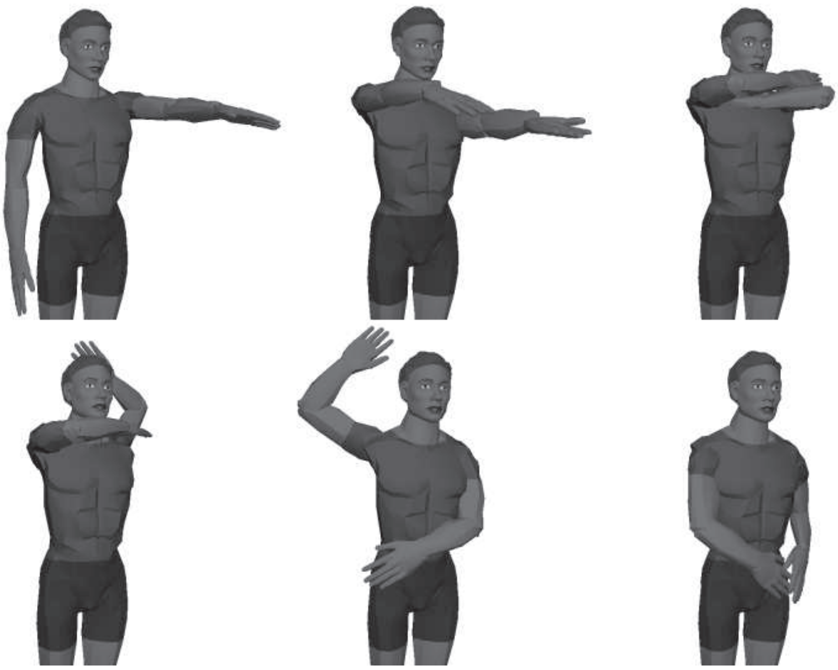

Figure 4. An example of Adonis performing the Macarena, shown as a series of snap-shots, in this case using the joint-space torque PD-servo controller.

impedance controller was also found to be less sensitive to singular configurations of the arms (such as in sub-task 1, where the arm is straight). For these reasons, we chose not to use this final control method for evaluation; for more details on this implementation, see Matarić, Zordan & Mason (1998*b*).

8. Performance Analysis and Comparisons

Analysis and evaluation of complex behavior is an open research challenge. As synthetic behaviors for agents in animation, robotics, and AI become more complex, the issue of analysis becomes increasingly acute. In this section, we explore several evaluation criteria, both qualitative and quantitative, and make observations about differences between the different controllers performing the same task, consistencies from task to task for a single controller, and similarities between human and synthetic motion.

8.1. Naturalness of Movement: Qualitative

Judging the naturalness of movement is an important aspect of both robotic and animation evaluation, but aesthetic judgment is difficult to quantify. Qualitative judgments of motion require real-time playbacks of recorded behaviors; for the three controllers we implemented, those are available from: *http://www-robotics.usc.edu/~ agents/macarena.html*

Figure 5 shows a time-lapse image for sub-task 10 with the goal of facilitating a qualitative comparison of the arm trajectory generated by each of the three controllers. The impedance controller is shown on the left, torque-field controller in the middle, and the PD-servo controller on the right. While the beginning and end postures are very similar for all three, and all paths are valid

Figure 5. A time-lapse image of sub-task 10, showing the trajectories the hand takes using the different controllers: impedance on the left, torque-field in the middle, and PD-servo on the right.

in that they avoid body collisions and unnatural postures, the paths themselves vary significantly. The motion generated by the PD-servo is smooth but contains an exaggerated curve, due to the joint-space spline interpolation between the chosen via points. The torque-field movement is also smooth, resulting from the Gaussian controllers. In contrast, the impedance controller motion is more jerky because its set-point moves along straight lines.

Many differences between human movement and that of our simulated agents are due to the underlying dynamics of our chosen test bed; the qualitative features caused by the limitations of the dynamic simulation must be separated from those dictated by the underlying controller. Rigid body simulation imposes limitations that cannot be overcome by control. For instance, Adonis's un-actuated spine necessarily appears stiff. Furthermore, dynamic simulation constrains motion to be physically plausible but not necessarily natural. For example, since the simulation does not constrain joint limits or avoid collisions, the controllers must handle these limitations directly. Because the controllers have no knowledge of body boundaries, avoiding self-collisions was accomplished through the user's choice of desired positions and/or angles, resulting in conservative, unnatural trajectories. This can be improved with direct collision prediction and avoidance, as well as by built-in joint limits. In contrast to limitations caused by the simulation, some qualitative differences are caused by the controllers directly. For example, the joint-space torque method interpolated postures with splines to smooth the resulting motion. It also included small head and hand movements that produce more natural appearance for the overall motion.

Qualitative differences between controllers are often aesthetic, and thus difficult to quantify. Some metrics, such as comfort, can be applied, but even those vary under different dynamics and involve some observer/performer bias. To avoid this problem, the next section addresses two approaches to a more quantitative evaluation of the controllers.

8.2. Naturalness of Movement: Quantitative

The whole arm path, analyzed qualitatively in the previous section, is still too complex to easily compare in a quantitative fashion without introducing external metrics. To focus, we consider only the end-effector motion, particularly the velocity and jerk of the dominant or active hand during individual sub-tasks. As a base-case or control in this analysis, we use hand positions recorded from a human performing the Macarena.

8.2.1. *Comparison of End-Effector Speed*

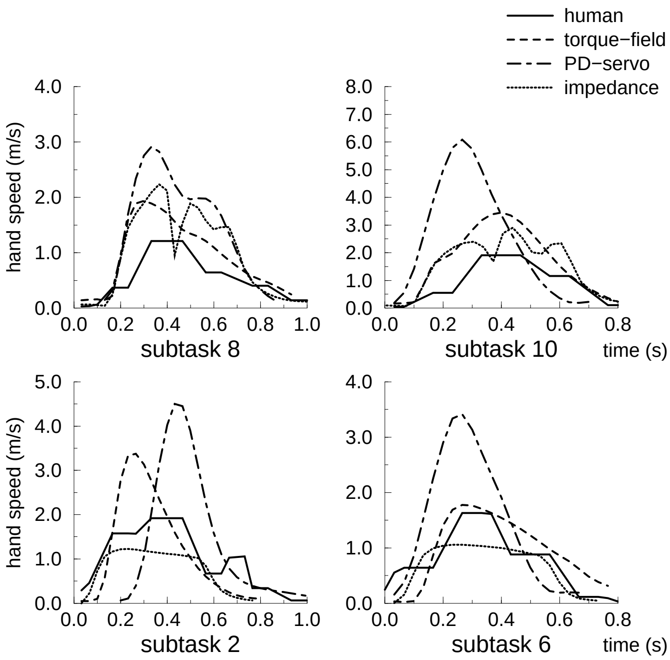

Figure 6. A comparison of the hand velocity profiles in four sub-tasks: sub-task 2 (extending the arm to straight out), sub-task 6 (moving from straight out to touching the shoulder), sub-task 8 (moving from shoulder to the back of the head), and sub-task 10 (moving from the back of the head to the ribs), and human data.

Hand position data of a person performing the Macarena were recorded with a commercial Flock of Birds electro-magnetic tracking system and used to compute the hand velocities. These are compared to the velocities of the three controllers we implemented; Figure 6 shows the velocities for the analyzed controllers and for a human performing the dance.

To evaluate an individual controller performing a given sub-task, we consider the overall shape and smoothness of the velocity profile as well as the peak speed. Since the human motion data was recorded at fairly low variable sample rates (about 5 samples/sec), it produces stair-step velocity profiles; we assume the effect would be smoothed with higher frequency samples. An analysis of peak velocities shows that the joint-space PD-servo torque controller generated unnaturally fast hand movements while the other two controllers more closely matched the human peak speeds. In contrast, the same controller generated the smoothest and most symmetric hand profiles; natural human movement has been categorized as having such symmetric properties (Morasso 1981, Atkeson & Hollerbach 1985). Furthermore, in the movements not requiring collision avoidance (sub-tasks 2 and 6), the impedance controller produced motion that closely matches the shape of the human velocity profile.

Differences in hand movements from task to task indicate how a controller performs over a variety of sub-tasks and suggest the potential generality of that controller for use in new tasks. Task variability exercises the controller by forcing it to perform in a variety of conditions. Notably, sub-tasks 8 and 10 require more sophisticated paths in order to avoid head/arm collisions. The PD-servo and impedance controllers use via points to avoid this collision. The effect of these postures can be seen most dramatically in the speed profile for sub-task 8, noting the change in speed corresponding to the posture change at about 0.5 seconds. However, the torque-field controller uses an initial pulse to control the overall trajectory and it remains more consistent across these tasks. Although the via points help achieve the goal of collision avoidance, the resulting velocity profiles indicate the need for a more sophisticated approach.

8.2.2. *Comparison of End-Effector Jerk*

Minimal jerk of hand position has been proposed by Flash & Hogan (1985) as a metric for describing human arm movements in the plane. Inspired by their work in planar motion, we propose a 3D evaluation metric, according to the following index:

$$C_j = \frac{1}{2}\int_0^{t_{final}} \frac{\partial^3 x}{\partial t^3} + \frac{\partial^3 y}{\partial t^3} + \frac{\partial^3 z}{\partial t^3} dt \tag{11}$$

where $\partial^3 x/\partial t^3$ is the third derivative of x, y and z positions with respect to time. We chose jerk as an evaluation metric over other measures such as minimum torque change (Uno, Kawato & Suzuki 1989) or energy (Nelson 1983), because it is much easier to record from a human subject and is also a good measure of smoothness.

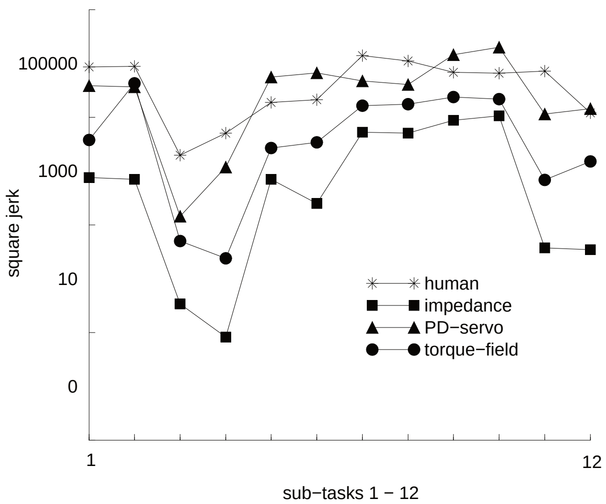

Figure 7. A comparison of the jerk values for the different controllers (PD-servo, torque-field, and impedance), and for human data. The lines connecting the data points do not correspond to actual data, since the sub-tasks are calculated independently, and map to left and right hand movements.

The calculated jerk values of the three different controllers and the human data are shown in the graph (Figure 7) corresponding to the square jerk for the active hand (e.g., in sub-task 1 the right arm, in sub-task 2 the left arm, and so on) over the length of the task. We do not expect a correspondence between the controllers and the human jerk values, but instead focus on trends across sub-tasks. As expected, movements that involve collision avoidance with the head (i.e., sub-tasks 7 through 10) have high jerk values overall, reflecting their complexity. Since jerk is based solely on Cartesian movement, it is low for movements that are primarily specified by joint constraints (i.e., sub-tasks 3 and 4 which command “turn the hand palm up”). Finally, low jerk also results from movements over short distances between Cartesian goals (i.e., sub-tasks 11 and 12, moving the hand from one hip to the other).

Jerk is a sensitive measure that varies strongly from task to task and from controller to controller; thus the log scale. Furthermore, the motion-capture system used to gather human data can suffer from marker slippage, adding further noise into the evaluation. We made no effort to create correspondence between the paths taken by the human and the different controllers, and thus variability in arm path is unaccounted for. Finally, timing has an effect on the jerk; slower movements have less jerk than faster ones. The movements shown do not all have the same timing and, although we tried various methods to normalize according to the timing, the data shown do not account for these differences explicitly, i.e., are not normalized. Therefore, the exact values in this graph are less reliable than the general trends they indicate, and it is remarkable to see the obvious correlations between the different data sets.

8.3. Controller Use and Flexibility

In addition to evaluating the success of the controllers in creating a life-like Macarena, we have also evaluated the controllers from the user's point of view. In this section we consider issues such as the amount of information required by each controller, the ease with which that information is input to the simulation, the simplicity with which the final motion is tuned for the various cases, and the actual computational complexity of the controllers themselves.

Once the gains and other constants have been fixed, there is not a great difference in the amount of information required by the three different controllers. The torque-field controller has the lowest overhead, requiring 14 values per arm per sub-task (7 for the step function, and 7 for the pulse). The PD-servo controller requires only 7 values per arm, but these need to be input at every time-step of the simulation, thus calling for an extra interpolation routine. The impedance controller also requires 7 values, including the hand position, orientation and the elbow orientation; like the PD-servo, it also uses an interpolation routine.

Rather more important than the number of parameters needed to specify a particular position is the ease with which that information is determined. For the PD-servo and torque-field controllers, this information is input in joint-space, so the user needs to solve the inverse kinematics of the arm manually, usually by trying different angles and adjusting. This is straight-forward if a little tedious, due to the fact that the joints are in an articulated chain, making the effect of any one joint on the arm motion dependent on the angles of all the others. The impedance controller works in Cartesian space, which makes the specification of hand positions much easier. Specifying the orientations of the elbow and hand is slightly more difficult, however, mainly due to the awkwardness of specifying three-dimensional rotations. This illustrates the fundamental tradeoff between the two types of control; the joint-space controllers are awkward to use but have explicit control over all the joints, while the Cartesian space controller is easier to use, but has less control over the individual degrees of freedom.

A third factor is the influence of the dynamics of the arm. While dancing the Macarena, the arm is moving quickly enough for dynamics to be important, making the choice of set-points, and particularly via points, quite important. For the torque-field controller, the pulse torque-field requires hand-tuning to create the motion, while for the other controllers, the positions of the via points requires hand-tuning. Since the motion of the arm is not wholly determined by the positions of these points, it is difficult to map from an error in the arm path to changes in a specific parameter. This difficulty is apparent in both reference frames, for the same reasons as described previously.

A final evaluation can be made in terms of the complexity of the implementation. The most computationally simple controller is the PD-servo method, followed closely by the torque-field controller. The impedance controller is considerably more complex, requiring a 7-by-6 and a 7-by-3 Jacobian matrix to be calculated at each time-step, as well as numerous vector operations for gravity compensation. However, this is still considerably less complex than any explicit inverse kinematics algorithm. The increased complexity of the impedance controller presents a trade-off in return for the ease of specifying positions in Cartesian space.

9. Continuing Work: The Combination Approach

The three controller implementations we presented all involve unavoidable tradeoffs, because each uses only a single, consistent approach to generating movement. However, different reference frames appear even in the simplest task specifications, resulting in unnatural and challenging transformations between the specification and the implementation. From the stand-point of the user, as well as the appearance of the final synthesized behavior, it would be preferable to have a means of flexibly combining the different control alternatives, so as to always utilize the approach most suited for a given task or sub-task. We are currently working on developing just such an approach to control.

Our approach is implemented within the behavior-based framework (Matarić 1997, Brooks 1991), which uses *behaviors* as abstractions for encapsulating low-level control details within each primitive. Consequently, we can implement generic primitives such as *get-posture* and *go-to-point* and parameterize them with the specific goals of each sub-task, as it is assigned. One of the benefits of the behavior decomposition is not only that there are different ways of structuring a given system (i.e., different types of controllers), but also that once a behavior decomposition is achieved, the specific behavior controllers can themselves vary, depending on the available sensors and effectors. For example, *get-posture* can be implemented with different types of joint-space controllers, and, analogously, *go-to-point* can use different Cartesian controllers, if desired. Furthermore, other behavior types can be added, such as an oscillator-based primitives for movements such as bouncing, waving, swinging, etc (Williamson 1998).

In an early demonstration of this approach, Matarić, Williamson, Demiris & Mohan (1998*a*) employed the notion of different types of motor primitives as behaviors to generate the same Macarena sub-tasks. There, the sub-tasks were assigned different types of controllers: PD-servo joint-space control for posture-related sub-tasks (such as sub-tasks 1 through 4), and impedance Cartesian control, for extrinsic or body-centered movements (such as sub-tasks 5 through 12). Our implementation executed each sub-task sequentially, thus eliminating interference between the different controllers. Besides sequencing, however, behaviors/primitives can also be co-activated, i.e., executed in parallel. For example, our implementation included an *avoid-collisions* primitive executed concurrently with any *get-posture* or *go-to-point* primitive, in order to generate safe, collision-free movement. Concurrent behavior combination is more complex than sequencing, however, and requires consistent output representations between the controllers being combined (Matarić 1997).

Using different types of primitives assumes that either the user or some intelligent automated method can subdivide the overall task into sub-tasks, and assign those to the most appropriate types of behaviors/primitives. We believe that these are not unreasonable assumptions. Human-generated specifications are typically sequential and presented in a step-wise fashion. Sub-task breaks can also be generated directly from observing movement, such as for example using zero-velocity breaks for each end-point. Automatically assigning sub-tasks to primitives is more complex; it could be coarsely approximated using parsing and key-word search of the textual task specification, which provides strong hints in the form of references body parts and joints.

In such a combination control system, individual behaviors may utilize different representations, coordinate frames, and underlying computation, but their use and performance can be seamlessly integrated by sequencing and co-activation. An effective means of encapsulating generic behaviors would also allow the integration of control schemes from different users. As complex articulated agents become more prevalent, such a modular approach to control could use its “open architecture” to combine the advantages of various successful approaches.

10. Conclusion

We have compared a set of three approaches for control of anthropomorphic agents, including PD-servo control, torque-field control, and impedance control, implemented on the same dynamic torso simulation, Adonis, and tested on the same Macarena sub-tasks. We compared the three controllers against one another and against human data, using qualitative and quantitative metrics, including naturalness of appearance, hand velocity and jerk, and controller use and flexibility.

To facilitate a realistic comparison, the controllers and the human data were generated independently. However, various techniques can be implemented to generate a closer fit between the data, if that is desired. Specifically, human hand positions could be used to select goal positions for the impedance controller. Similarly, an IK solver could be used to compute postures for the joint-space controllers that achieve these hand positions. Timing taken from human motion could be used to generate simulated motion that more closely fits the human performance. Lastly, minimization techniques could be applied to the controller parameters to find movements that minimize jerk and/or match other performance metrics.

The fundamental tradeoff between believability and control effort still remains, as the approaches produce different results depending on sub-task specification. In order to address these tradeoffs, we proposed a combination framework which allows different types of movement primitives (under different reference frames and representations) to be used for different types of sub-tasks, in order to maximize the match between the description of the task and the controller that achieves it.

Acknowledgments

This work is supported by the NSF Career Grant IRI-9624237 to M. Matarić. The authors thank Nancy Pollard for help with the jerk calculations, Stefan Schaal and Jessica Hodgins for sharing expertise and providing many insightful comments. The Adonis simulation was developed by Jessica Hodgins at Georgia Institute of Technology.

References

Atkeson, C. G. (1989), 'Learning Arm Kinematics and Dynamics', *Annual Review of Neuroscience* **12**, 157–183.

Atkeson, C. G. & Hollerbach, J. M. (1985), 'Kinematic Features of Unrestrained Vertical Arm Movements', *Journal of Neuroscience* **5**, 2318–2330.

Badler, N. I., Barsky, B. A. & Zeltzer, D., eds (1991), *Making Them Move: Mechanics, Control and Animation of Articulated Figures*, Morgan Kaufmann Publishers, San Mateo, California.

Baker, D. R. & Wampler II, C. W. (1988), 'On the Inverse Kinematics of Redundant Manipulators', *International Journal or Robotics Research* **7**(2), 3–21.

Beccari, G. & Stramigioli, S. (1998), Impedance Control as Merging Mechanism for a Behaviour-Based Architecture, *in* 'Proceedings of the 1998 IEEE International Conference on Robotics and Automation', Leuven, Belgium, pp. 1429–1434.

Brady, M., Hollerbach, J. M., Johnson, T. L., Lozano-Perez, T. & Mason, M. T., eds (1982), *Robot Motion: Planning and Control*, MIT Press, Cambridge, MA.

Brooks, R. A. (1991), Intelligence Without Reason, *in* 'Proceedings, IJCAI-91', Sydney, Australia, pp. 569–595.

Cohen, M. F. (1992), Interactive Spacetime Control for Animation, *in* 'Computer Graphics (Proceedings, SIGGRAPH '92)', pp. 293–301.

Colgate, J. E. & Hogan, N. (1988), 'Robust control of dynamically interacting systems', *International Journal of Control* **48**(1), 65–88.

Craig, J. J. (1989), *Introduction to Robotics: Mechanics and Control*, second edn, Addison-Wesley, Reading, Massachusetts.

Flash, T. & Hogan, N. (1985), 'The coordination of the arm movements: an experimentally confirmed mathematical model', *Journal of Neuroscience* **7**, 1688–1703.

Giszter, S. F., Mussa-Ivaldi, F. A. & Bizzi, E. (1993), 'Convergent force fields organized in the frog's spinal cord', *Journal of Neuroscience* **13**(2), 467–491.

Hodgins, J. K., Wooten, W. L., Brogan, D. C. & O'Brien, J. F. (1995), Animating Human Athletics, *in* 'Computer Graphics (Proceedings, SIGGRAPH '95)', Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 71–78.

Hogan, N. (1985), 'Impedance Control: An Approach to Manipulation', *Journal of Dynamic Systems, Measurement, and Control* **107**, 1–24.

Hollars, M., Rosenthal, D. & Sherman, M. (1991), SD/Fast User's Manual, Technical report, Symbolic Dynamics, Inc.

Jordan, M. I. & Rumelhart, D. E. (1992), 'Forward Models: supervised learning with a distal teacher', *Cognitive Science* **16**, 307–354.

Klein, C. A. & Huang, C. (1983), 'Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators', *IEEE Transactions on Systems, Man and Cybernetics* **SMC-13**(3), 245–250.

Luh, J., Walker, M. & Paul, R. (1980), 'On-line computational scheme for mechanical manipulators', *Transactions ASME Journal of Dynamic Systems, Measurements and Control* **120**, 69–76.

Marjanović, M., Scassellati, B. & Williamson, M. (1996), Self-Taught Visually-Guided Pointing for a Humanoid Robot, *in* P. Maes, M. Matarić, J.-A. Meyer, J. Pollack & S. Wilson, eds, 'Fourth International Conference on Simulation of Adaptive Behavior', The MIT Press, Cape Cod, MA, pp. 35–44.

Matarić, M. & Pomplun, M. (1997), What do People Look at When Watching Human Movement?, Technical Report CS-97-194, Brandeis University.

Matarić, M. J. (1995), 'Designing and Understanding Adaptive Group Behavior', *Adaptive Behavior* **4**(1), 50–81.

Matarić, M. J. (1997), 'Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior', *Journal of Experimental and Theoretical Artificial Intelligence* **9**(2–3), 323–336.

Matarić, M. J. & Pomplun, M. (1998), 'Fixation Behavior in Observation and Imitation of Human Movement', *to appear in Cognitive Brain Research.*

Matarić, M. J., Williamson, M. M., Demiris, J. & Mohan, A. (1998*a*), Behavior-Based Primitives for Articulated Control, *in* R. Pfiefer, B. Blumberg, J.-A. Meyer & S. W. Wilson, eds, 'Fifth International Conference on Simulation of Adaptive Behavior', MIT Press, Cambridge, MA, pp. 165–170.

Matarić, M. J., Zordan, V. B. & Mason, Z. (1998*b*), Movement Control Methods for Complex, Dynamically Simulated Agents: Adonis Dances the Macarena, *in* 'Autonomous Agents', ACM Press, Minneapolis, St. Paul, MI, pp. 317–324.

Morasso, P. (1981), 'Spatial control of arm movements', *Experimental Brain Research* **42**, 223–227.

Mussa-Ivaldi, F. A. (1997), Nonlinear force Fields: A Distributed System of Control Primitives for Representing and Learning Movements, *in* 'Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation', Monterey, CA.

Mussa-Ivaldi, F. A. & Giszter, S. F. (1992), 'Vector field approximations: a computational paradigm for motor control and learning', *Biological Cybernetics* **67**, 491–500.

Mussa-Ivaldi, F. A., Giszter, S. F. & Bizzi, E. (1994), 'Linear combinations of primitives in vertebrate motor control', *Proceedings of the National Academy of Sciences* **91**, 7534–7538.

Nelson, W. L. (1983), 'Physical Principles for Economies of Skilled Movements', *Biological Cybernetics* **46**, 135–147.

Ngo, J. T. & Marks, J. (1993), Spacetime Constraints Revisited, *in* J. T. Kajiya, ed., 'Computer Graphics (Proceedings, SIGGRAPH '93)', pp. 343–350.

Pai, D. (1990), Programming Anthropoid Walking: Control and Simulation, Technical Report Computer Science Tech Report TR 90-1178, Cornell University.

Paul, R. P. (1981), *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA.

Raibert, M. & Hodgins, J. (1991), Animation of Dynamic Legged Locomotion, *in* 'Computer Graphics (Proceedings, SIGGRAPH '91)', pp. 349–356.

Raibert, M. H. & Craig, J. J. (1981), 'Hybrid Position/Force Control of Manipulators', *Journal of Dynamic Systems, Measurement, and Control* **103**, 126–133.

Salisbury, J. K. (1980), Active Stiffness Control of a Manipulator in Cartesian Coordinates, *in* 'Proceedings or the 19th IEEE Conference on Decision and Control', pp. 95–100.

Schaal, S. (1997), Learning from demonstration, *in* M. Mozer, M. Jordan & T. Petsche, eds, 'Advances in Neural Information Processing Systems 9', The MIT Press, pp. 1040–1046.

Schaal, S. & Atkeson, C. C. (1994), 'Robot Juggling: An Implementation of Memory-Based Learning', *Control Systems Magazine* **14**, 57–71.

Slotine, J.-J. E. & Li, W. (1991), *Applied nonlinear control*, Prentice Hall, Englewood Cliffs, N.J.

Uno, Y., Kawato, M. & Suzuki, R. (1989), 'Formation and Control of Optimal Trajetory in Human Arm Movement-Minimum Torque-Change Model', *Biological Cybernetics* **61**, 89–101.

van de Panne, M. & Lamouret, A. (1995), Guided Optimization for Balanced Locomotion, *in* D. Terzopoulos & D. Thalmann, eds, 'Computer Animation and Simulation '95', Eurographics, Springer-Verlag, pp. 165–177. ISBN 3-211-82738-2.

Whitney, D. E. (1969), 'Resolved motion rate control of manipulators and human prostheses', *IEEE Transactions on Man-Machine Systems* **10**(2), 47–53.

Whitney, D. E. (1982), The mathematics of coordinated control of prosthetic arms and manipulators, *in* M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Pérez & M. T. Mason, eds, 'Robot Motion: planning and control', MIT Press, pp. 287–304.

Williamson, M. (1996), Postural Primitives: Interactive Behavior for a Humanoid Robot Arm, *in* P. Maes, M. Matarić, J.-A. Meyer, J. Pollack & S. Wilson, eds, 'Fourth International Conference on Simulation of Adaptive Behavior', The MIT Press, Cape Cod, MA, pp. 124–131.

Williamson, M. M. (1998), Rhythmic robot control using oscillators, *in* 'Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98)'.

Witkin, A. & Kass, M. (1988), Spacetime Constraints, *in* 'Computer Graphics (Proceedings, SIGGRAPH '88)', pp. 159–168.