Stefan Schaal[†] **and Dagmar Sternad**[‡]
[†]Department of Brain and Cognitive Sciences and the Artificial Intelligence Laboratory, MIT, 545 Technology Square, Cambridge, MA, 02139 (Email: sschaal@ai.mit.edu);
[‡]Center for the Ecological Study of Perception and Action, University of Connecticut, 406 Babbidge Road, Storrs, CT, 06268

Learning of Passive Motor Control Strategies with Genetic Algorithms

Abstract: This study investigates learning passive motor control strategies. Passive control is understood as control without active error correction; the movement is stabilized by particular properties of the controlling dynamics. We analyze the task of juggling a ball on a racket. An approximation to the optimal solution of the task is derived by means of optimization theory. In order to model the learning process, the problem is coded for a genetic algorithm in representations without sensory or with sensory information. For all representations the genetic algorithm is able to find passive control strategies, but learning speed and the quality of the outcome are significantly different. A comparison with data from human subjects shows that humans seem to apply yet different movement strategies to the ones proposed. For the feedback representation some implications arise for learning from demonstration.

1. Introduction

Despite all advances in research on human motor coordination and motor learning, little understanding has been gained so far how these skills are accomplished. From the perspective of control theory[5], two major control approaches are distinguished. *Closed loop control* requires continuous sensing of the current state of the system; if the planned state differs from the actually achieved one, a modification of the next actuator command compensates for this error. In *open loop control*, on the other hand, the spatio-temporal sequence of actuator commands is determined before the movement starts and then executed to plan. Since no feedback is provided, there is no possibility of error correction. While closed loop control can be considered (re-)active control, open loop control is essentially passive.

The central nervous system (CNS) possesses two control circuits with resembling properties. On the spinal level, a short feedback loop takes care of fast movement regulation. Although feedback is involved, this control receives no input of cortical areas and, in a figurative sense, can be considered passive. In addition to this low level regulation, higher brain areas may influence the low level circuitry at any time via long feedback loops through the cortical motor centers.

Passive control is appealing because of its low computational load during movement. One could imagine that for movement initiation a "control package" were delivered

In: Nadel, L., & Stein, D. (Eds.), *1992 Lectures in complex systems* (pp.631-643).
Redwood City, CA: Addison-Wesley.

to the spinal level which triggers an autonomous control circuit to sustain the movement afterwards. The higher brain functions would be free for other tasks and only check for correctness and stability of the movement at discrete events. Some evidence for the biological plausibility of such control procedures has been shown in the work about central pattern generators[6].

We investigate the cyclic movement of juggling a ball on a paddle to find out whether some form of passive control can be learned. An approximation of the optimal solution of the task in section 2 provides an evaluation criterion for a series of learning experiments with genetic algorithms presented in section 3. Section 4 discusses the results of the experiments and compares them to empirical data from human subjects.

2. Analysis of Paddle Juggling

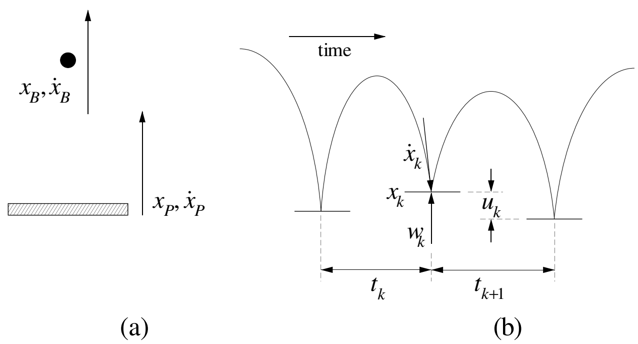

FIGURE 1 (a) Sketch of the paddle juggling setup; (b) notation for the discretization

Figure 1 displays the setup and notation of paddle juggling. The ball bounces on the paddle under gravity, and the movement of the paddle tries to sustain a regular bouncing motion. A coefficient of restitution $\alpha \in [0,1]$ models the elastic ball-paddle impact. The system can be discretized with a Poincaré section $\Sigma = \left\{ \left(\mathbf{x_B}, \mathbf{x_P}\right) \in \Re^4 | x_B - x_P = 0 \right\}$. With the notation of Figure 1b, where $(\dot{x}_k)$ denotes ball velocity and (x_k) ball position immediately before the impact where the paddle velocity is (w_k), and where (u_k) denotes the vertical position shift between consecutive impacts, the discrete system equations are:

$$
\begin{aligned}
\dot{x}_{k+1} &= -\sqrt{\left((1+\alpha) w_k - \alpha \dot{x}_k\right)^2 - 2 g u_k}, \\
x_{k+1} &= x_k + u_k, \\
t_{k+1} &= \frac{1}{g}\left(\left((1+\alpha) w_k - \alpha \dot{x}_k\right) + \dot{x}_{k+1}\right).
\end{aligned}
\tag{1}
$$

Paddle juggling was investigated by several groups in recent years. For the vibrating paddle (high oscillation frequency with small amplitude) it could be shown analytically and experimentally that the system exhibits period bifurcations, strange attractors, and chaos-like motion[7]. Systems which had to control and learn this task (at a moderate juggling frequency) were examined in robotics[1,2,8,9]. Depending on the control algorithm, the execution of this task does not necessarily need feedback: driving the paddle with a sinusoidal motion results in a dynamic system which has a trapping region and exhibits stable bouncing patterns under certain parameter settings,

2.1 Paddle Juggling as an Optimization Problem

Paddle juggling can be formulated as an optimization problem. For these calculations it will be assumed that the setpoint, at which the ball shall be juggled, is a given impact state $\mathbf{x}_s = (\dot{x}_s, x_s)^T$, whose setpoint controls $\mathbf{u}_s = (w_s, u_s)$ result implicitly from a periodic paddle trajectory $(\mathbf{x}_P(t))$. The setpoint is entirely determined by one parameter of the juggling motion, which can be either the period (τ), the maximal ball height (h), or the impact velocity $(\dot{x}_S)$. In this regulator problem, the task of the controller is to keep the system at the setpoint. If a perturbation displaces the ball from its setpoint, it has to be guided back. By modeling the paddle motion as an r*th*-order Fourier series:

$$x_P(t) = \frac{a_0}{2} + \sum_{i=1}^{r} a_r \cos(r\omega t) + b_r \sin(r\omega t) \tag{2}$$

a multistage optimization problem[3] is formed, subject to minimizing the cost function:

$$J = \phi(\mathbf{x}_n, \mathbf{p}) + \sum_{k=0}^{n-1} L(\mathbf{x}_k) \quad \text{, where}$$

$$\phi(\mathbf{x}_n, \mathbf{p}) = (\mathbf{x}_n - \mathbf{x_s})^T \Phi\, (\mathbf{x}_n - \mathbf{x_s}) + c\frac{1}{2}\sum_{i=1}^{r} (i\omega)^6 (a_i^2 + b_i^2), \tag{3}$$

$$L(\mathbf{x}_k) = (\mathbf{x}_k - \mathbf{x}_s)^T \mathbf{Q}\, (\mathbf{x}_k - \mathbf{x}_s)^T, \qquad \omega = \frac{2\pi}{\tau}.$$

$(\mathbf{x}_k)$ denotes the ball state vector at stage (k) and the matrices $(\Phi, \mathbf{Q}, \mathbf{R})$ are weight matrices. The last term of the equation for terminal cost (ϕ) in (3) represents a so-called jerk term and is weighted by the factor (c). Jerk denotes the third derivative $(\dddot{x}_P)$ of paddle position with respect to time, imposing a biologically motivated smoothness constraint on the paddle acceleration[4]. Without this term in the cost function, any optimization would minimize deviations from the setpoint with unrealistically sharp movements. In sum, the formalism of (3) tries to guide the perturbed ball smoothly back to the setpoint in an n-stage sequence: given the initial conditions of the perturbed ball $\mathbf{x}_{k=0} = (\dot{x}_0, x_0)^T$, the Fourier coefficients $\mathbf{p} = (a_0, a_1, \ldots, a_r, b_1, b_2, \ldots, b_r)^T$ are to be calculated such that Eq. (3) is minimized. Note that (u_k) and (w_k) are not present in (3), meaning that the paddle trajectory is independent of the ball motion; stability can only come from an appropriate choice of the Fourier coefficients. The system is thus passively controlled.

The numerical solution of (3) was obtained with the gradient method of dynamic programming[3]. The optimization was done for a 10-stage task, a fifth-order Fourier series, and a coefficient of restitution $\alpha = 0.7$. Initial conditions yielded:

$$\begin{aligned}&\dot{x}_0 \in \left[-0.12\ \dot{x}_S, +0.12\ \dot{x}_S\right]\\ &x_0 \quad \text{such that} \quad t_{init} \in \left[-0.12\ \tau, +0.12\ \tau\right], \quad \text{where} \quad t_{init} = \tau - t_0\end{aligned} \tag{4}$$

and the weight matrices $(\Phi, \mathbf{Q})$ were chosen according to a common heuristic:

$$\mathbf{Q} = \frac{1}{100}\Phi = \begin{pmatrix} 1/\dot{x}_{\max}^2 & 0 \\ 0 & 1/x_{\max}^2 \end{pmatrix} \tag{5}$$

The values $(\dot{x}_{\max}, x_{\max})$ correspond to the maximal ranges of the intervals in Eq. (4). A relatively large weight of (Φ) makes sure that the ball comes back to the setpoint. A moderate choice of (c) in Eq. (3), in order to make the penalty of jerk terms to be roughly one forth of the total cost, resulted in a paddle motion which was close to a sinusoid but still had some power in the higher harmonics. Figure 2 shows one result in comparison with a sine function in state space. Linear stability analysis of Eq.(1) modified by Eq.(2) holds that for stable juggling the ball must be hit in the first quadrant of Figure 2.

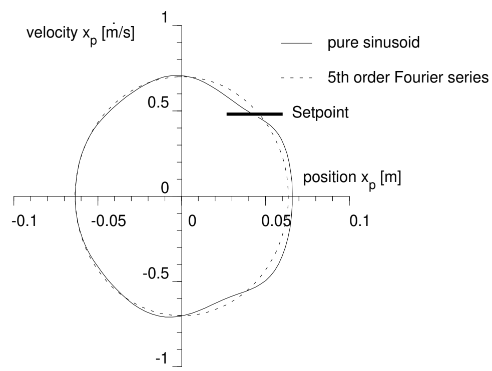

FIGURE 2 State space plot of an optimal paddle trajectory (period $\tau = 0.57\,\text{sec}$)

3. Learning Experiments with Genetic Algorithms

The following numerical experiments will explore genetic algorithms (GA) (cf. contribution of M. Mitchell in this book) to simulate a reinforcement learning process. Similar to optimization analysis, reinforcement learning requires a performance index to evaluate the quality of the outcome. In order to apply genetic algorithms to paddle juggling, the task must be encoded as a gene-string, and an appropriate fitness function has to be found.

In a first approach to the problem, the periodic paddle movement is divided into a set of twenty real-valued position values (Figure 3), represented by the first twenty genes in the gene-string. The 21st gene codes the period (τ), the 22nd gene a scaling factor which multiplies each of the twenty position genes. Given a population of randomized initial genomes, the task of the GA is to find a sequence of position values to perform a paddle movement by which the ball is juggled in a stable fashion. The scaling factor allows to stretch the movement in the spatial dimension, the period value is the corresponding temporal stretch factor.

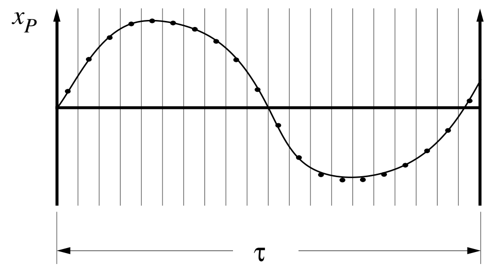

FIGURE 3 Discretized representation of periodic paddle movement for genetic algorithm

The performance (or fitness) of a paddle trajectory is determined by:

$$J = \frac{10.0}{\bar{j} + 5\frac{\sigma_h}{\bar{h}} + 5\frac{\sigma_p}{\bar{\tau}_b} + 0.8\frac{\tau}{\bar{\tau}_b}} \qquad (6)$$

where $(\bar{j})$ denotes the mean jerk per period, (σ_h, σ_p) the standard deviation of maximal height of the ball and the standard deviation of the bounce period, respectively, and $(\bar{h}, \bar{\tau}_b)$ the mean maximal height and mean bounce period of the ball, respectively. The distinction between bounce period $(\bar{\tau}_b)$ and period (τ) is necessary because the ball can bounce several times on the paddle during one paddle period (τ). The cost function J penalizes jerk, irregular juggling height, irregular bouncing period, and, with the last term in the denominator, a large number of bounces during one period (τ). By taking the inverse of the cost function, the minimization problem of section 2 becomes a maximization task in accordance with the usual GA formulation.

The statistical parameters of Eq. (6) were derived by submitting the paddle trajectory of each genome to a paddle juggling simulation. The discrete trajectory values were treated as the desired position values $x_{P(desired)}$ of a PD controller[5]. By differentiating this trajectory with respect to time, the desired velocities $\dot{x}_{P(desired)}$ for PD control at each discrete time event were derived. With this information and the specifications of the paddle mass, the PD controller is able to generate a smooth pursuit of the encoded trajectory if the trajectory is smooth enough. The paddle movement had a limited workspace of ±0.5m, the maximal acceleration of the paddle was restricted to 60m/s^2, the paddle mass was 0.5kg, and the ball's coefficient of restitution was $\alpha = 0.7$. The position and velocity gain was set to a constant value of $K_P = 200$ and $K_V = 20$. At the start of the simulation, the ball was dropped from 0.4m above the paddle. After a transient time of roughly five periods, the statistical values of Eq. (6) were derived from 20 subsequent periods. All GA experiments had a population size of 100 genomes, mutation probability $P_{mut} = 0.01$ per gene, crossover probability $P_{cross} = 0.8$ using double-crossover, and a proportional offspring reproduction mechanism allowing at most 5 offspring for the best genome.

3.1. Representation without Perception

In the setup of the first experiment all genomes where randomized within a reasonable range of the individual genes. After about 400 to 800 generations, the GA solution converged to a steady value. Figure 4a shows the phase portrait of the best result that the algorithm developed.

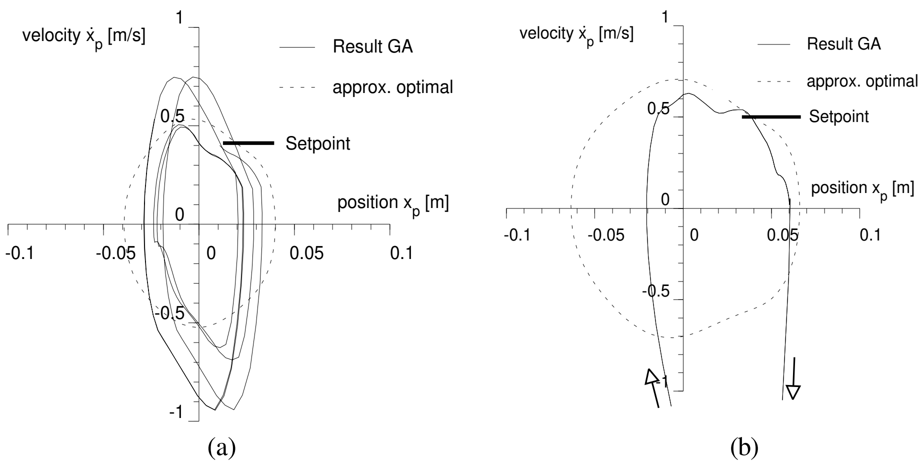

FIGURE 4 Outcome of learning without feedback: (a) best result of absolute coding (J=3.8); (b) best result of relative coding (J=4.2)

Each chart in Figure 4 contains a segment of a paddle trajectory over three successive periods after the transient time had elapsed. To enable an assessment of the solutions, the mean period $(\overline{\tau}_b)$ was used to calculate an approximately optimal paddle trajectory for each GA outcome. Different position coordinates of the setpoints were adjusted for to make the setpoint of optimal solution and GA solution align as well as possible; a shift in position coordinate corresponds to a redefinition of the reference coordinate system and does not change the results. The ideal paddle trajectory in phase space should be a smooth cycle. If the trajectories of successive periods traverse in the graph, the PD-controller was not able to follow the discrete trajectory plan, i.e., the encoded trajectory was not smooth enough. This is rather pronounced in the solution of the absolute coding (Figure 4a). Such a juggling pattern would be vulnerable to perturbations.

The first coding of the paddle trajectory was based on the assumption that a movement plan is be based on a discrete position representation of one period. An alternative representation would be in relative coordinates and the movement could be described as an "up-up-up-up-down-down-down-down-up-up-up-…" plan. In order to find out whether such relative position representation is more suitable, the second simulation encoded the relative change of position from step to step. The genes were still continuously valued, only the permissible range of their values was decreased appropriately. The spatial and temporal scaling as described above remained unchanged. An exemplary result is displayed in Figure 4b. This trajectory applied a “one-leave-out” strategy which can be inferred from the high negative velocity of the trajectory in 4b (which is partly clipped). Instead of hitting the ball in every cycle, it hit it only every other cycle. Although this might be considered cheating, it is a valid solution to the given problem and is particularly rewarded by the last term in the denominator of the cost function. Therefore, the seemingly high fitness of J=4.6 does not reflect the real quality of this trajectory and has to be corrected to J=3.7. On the whole, this representation achieved a significantly faster speed of learning (200 to 300 generations) as well as a higher maximum fitness.

3.2. Representation with Visual Perception

The third experiment addressed the question to what extend perception improves the speed of learning and the quality of the outcome, and, in particular, whether with perception the passively stable control strategy of the other experiments can still be found.

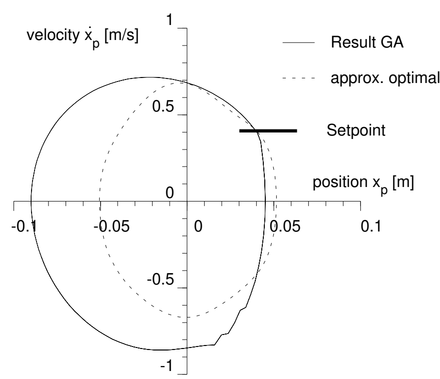

FIGURE 6 Best result of feedback GA (J=4.6) (for the optimal trajectory J>4.8)

To address this question, the representation of the problem had to be changed again. It was assumed that in visual perception the absolute position (x_B) of the ball, its velocity $(\dot{x}_B)$, and the ball’s relative distance to the paddle (x_R) can be perceived. These terms are multiplied by appropriate coefficients and summed up to specify the next desired paddle position and the next desired paddle velocity:

$$\dot{x}_{P_{k+1}} = (c_{11}\dot{x}_{B_k} + c_{12}x_{B_k} + c_{13}x_{R_k}) \cdot s$$
$$x_{P_{k+1}} = (c_{21}\dot{x}_{B_k} + c_{22}x_{B_k} + c_{23}x_{R_k}) \cdot s \qquad (7)$$

The task of the GA was to optimize the “ball-paddle coupling factors” (c_{ij}) and the scaling factor (s). This type of coding only requires seven continuously valued genes. In contrast to the previous representations, the ordering of the genes in the genomes was no longer essential.

Figure 6 depicts the best result out of five trials; the other trials were qualitatively the same. In the crucial impact region (1st quadrant) the GA solution comes very close to what was calculated as the optimal solution, as well as its entire fitness evaluation of J=4.6 differs only marginally from the score of the optimal with J>4.8. The optimal trajectory could score better if its initial phase was perfectly adjusted to minimize the transient time to reach the setpoint. Learning speed was significantly improved by using this representation. Within 10 to 30 generations very good trajectories were accomplished.

4. Discussion

The goal of this paper was to study learning of passive motor control strategies by using genetic algorithms. Passive control strategies do not need continuous re-planning of the movement in order to compensate for perturbations during the movement task, but accomplish the task by relying on a self-organized stabilization due to special properties of the control method or other parameters[9]. Three different representations of the task to juggle a ball on a paddle were compared with respect to their learning speed and quality of solutions, in particular, whether a passive control strategy could be found.

4.1. Simulation Results

The results of all three setups were positive in that they converged onto a passive control regime. However, learning speed and the quality of the outcome differed substantially between the three conditions. Figure 7 illustrates this by showing the mean population fitness of representative runs as a function of the number of generations. The relatively poor results of the absolute position coding in the GA are not surprising. This kind of coding had the most degrees of freedom by being able to jump from one extreme position to another in every single time step. The initial trajectory coded by the randomized genomes looks like a zig-zag line. Since GAs do not make use of local information as given by gradients, it takes many iterations until the zig-zag line is smoothed to a trajectory fit to perform juggling. The likelihood that the algorithm converges to a local minimum is large. Relative position encoding, on the other hand, does not tend to have the same jaggedness as absolute coding, because the jump from time step to time step is confined to a rather small range. The crossover operator in GAs will also do less harm to a relative position coding. Swapping an absolute position gene with another genome will always be detrimental if this gene codes anything else but the position actually needed. In contrast, exchanging a relative position gene will not destroy too much as long as the gene retains an appropriate “up” or “down” information. The relative position genes also facilitate the formation of building blocks. The essence of successful juggling is an “up-up-up” sequence in the impact area of the ball. Via crossover, this generally valid building block can easily be tested in several places of the paddle trajectory. A building block containing absolute position information, on the other hand, is unlikely to be useful in other genomes; it is difficult to smoothly integrate it into the already existing genes. In sum, if a genome shall represent a smooth function, relative coding seems to be of advantage.

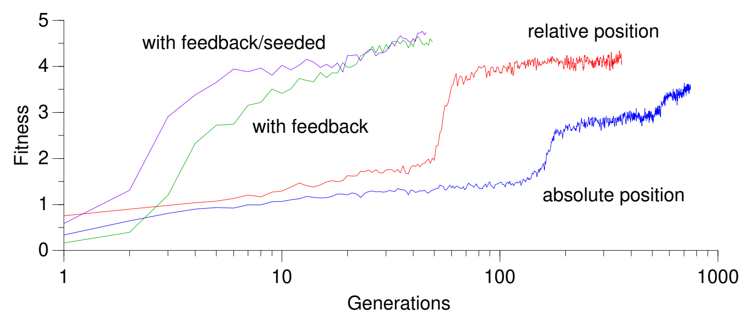

FIGURE 7 Comparison of learning speed of the three different representation

The quality of the results and of the learning speed was unexpectedly good for the representation with perception. Whereas learning speed must necessarily improve due to the comparatively short genomes, the almost optimal outcome was by no means self-evident. A particularly interesting property of the resultant paddle trajectory is that it still possesses the major characteristics of a passive juggling strategy, i.e., the impact takes place while the paddle position still increases and the velocity decreases. This passive stability property even allows to sustain stable juggling when perception is cut off (this requires a change to an open loop control algorithm). Apparently, movement learning can profit from active control. If the learned control scheme also allowed passive control, the system could gradually switch from active to passive control at an advanced level; mechanisms for that shall not be considered here. This seems to be plausible from monitoring the attention humans devote to a task in the learning and in the skilled stage.

4.2. Comparison with Human Juggling

To investigate the biological plausibility of the simulation results, the trajectory of the feedback GA was compared with data collected from an experiment in which a human subject juggled a tennis ball on a tennis racket. The phase plot in Figure 8a shows an average phase plot over a 30sec long run. Since the coefficient of restitution was different from that used in the GA simulations, the phase plots in Figure 8a can only serve for qualitative comparison.

As the most noticeable difference in human juggling, the balls impacts with the racket shortly after the positive paddle velocity peak while the optimal result as well as the GA feedback result have the setpoint far more in the declining part of the trajectory. Thus, the human juggling data stays very close to the limit of passive stability, and it cannot be resolved whether this juggling strategy is more on the passive or active control side. So far, not enough data has been collected from human subjects to allow any generalization. One reason why the simulations did not produce a juggling strategy more similar to humans may be that the dynamic and kinematic properties of the human arm where not taken into account. Another can be that the chosen performance criterion was inadequate.

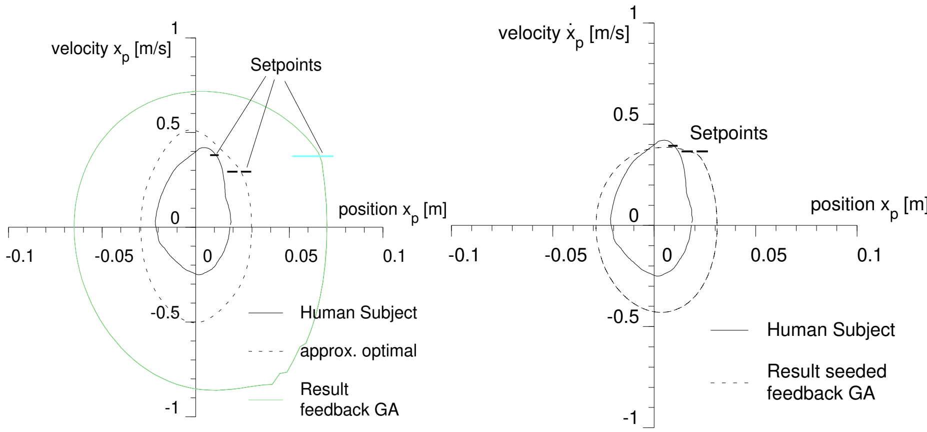

FIGURE 8 Comparison of results with data from human subject

An intriguing advantage of human motor learning over machine learning is due the fact that humans do not start out absolutely “uninitiated”. Firstly, past experience seems to play a non-negligible role. Secondly, humans often seem to extract an idea how to approach the task by watching somebody else’s performance, reading a “how-to” book, etc. These sources can partially specify the initial strategy and avoid long and fruitless experimentation. An important question therefore is what the learner extracts from a demonstration. The successful performance of the feedback GA may suggest that picking up something like coupling coefficients could be advantageous. Such a notion of coupling between perception and action could also be a promising route to transfer knowledge between different tasks. To test this hypothesis, the coupling coefficients of Eq. (7) were regressed from the human data, and the feedback GA was seeded with random variations of these coefficients. The result is shown in Figure 7 and Figure 8b. As can be seen, the GA solutions converged faster, although, as discussed before, human juggling produced different results to the simulations, and although the properties of tennis ball and tennis racket were different to the ones in the simulation. Now, the juggling setpoint lies closer to the peak velocity which is more similar to the human data. It needs to be mentioned, however, that the unperturbed regression data could not be used for the juggling simulation right away. It resulted in a paddle movement which was too large and continuously increasing.

Summarizing, this study showed once again the importance of how a motor task is represented in terms of its structure and variables. Different representations change learning speed and quality of results dramatically. So far we have little knowledge, how appropriate representations can be developed in order to obtain more insight into what it is that should be learned.

Acknowledgments

The work of this paper was supported by grants of the German National Scholarship Foundation to each of the authors, a fellowship of the Alexander von Humboldt Foundation to the first author, an NSF Grant BNS-9109880 awarded to M. T. Turvey, University of Connecticut, an Air Force Office of Scientific Research grant AFOSR-89-05000 and a National Science Foundation Presidential Young Investigator Award awarded to Christopher G. Atkeson, MIT.

References

1. Aboaf, E.W.; Drucker, S.M.; Atkeson, C.G.: Task-Level Robot Learing: Juggling a Tennis Ball More Accurately. *Proceedings of IEEE Interational Conference on Robotics and Automation*, May 14-19, Scottsdale, Arizona (1989).

2. Bühler, M.: *Robotic Tasks with Intermittent Dynamics*. Dissertation at Yale University (1990).

3. Dyer, P.; McReynolds, S.R.: *The Computation and Theory of Opitmal Control*. New York: Academic Press (1970).

4. Flash, T.; Hogan, N. : The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. *The Journal of Neurosience*, vol.5, no.7, pp.1688-1703 (1985).

5. Friedland, B. : *Control System Design: An Introduction to State-Space Methods*. New York: McGraw-Hill (1986).

6. Getting, P.A. : Comparative Analysis of Invertebrate Central Pattern Generators. *Neural Control of Rhythmic Movements in Vertebrates*, Cohon, H.; Rossignol, S.; Grillner, S.T. (eds). New York: Wiley (1988).

7. Guggenheimer, J.; Holmes, P.: *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York: Springer (1983).

8. Rizzi, A.A.; Koditschek, D.E. : Progress in Spacial Robot Juggling. *Proceedings of IEEE International Conference on Robotics and Automation*, Nice, France (1992).

9. Schaal, S.; Atkeson, C.G.; Botros, S. : What Should Be Learned?. In: *Proceedings of Seventh Yale Workshop on Adaptive and Learning Systems*, pp.199-204 (1992).