Nonlinear Dynamical Systems for Imitation with Humanoid Robots

Auke Jan Ijspeert[1,2] & Jun Nakanishi[1] & Tomohiro Shibata[2] & Stefan Schaal[1,2]
[1]Computational Learning and Motor Control Laboratory
University of Southern California, Los Angeles, CA 90089-2520, USA
[2]Kawato Dynamic Brain project, ATR, Kyoto 619-0288, Japan
Email: ijspeert@usc.edu, nakanisi@rubens.usc.edu, tom@erato.atr.co.jp, sschaal@usc.edu

Abstract

We present control policies (CPs) based on nonlinear differential equations with well-defined attractor dynamics for learning by imitation and trajectory formation in humanoid robotics. The CPs can fit complex movements (presented as joint-angle trajectories) using incremental locally-weighted regression techniques. Being implemented as autonomous nonlinear differential equations gives the CPs interesting properties such as robustness against strong external perturbations and the ability to generate movements which can easily be modified by additional perceptual variables and re-used for new targets.

We evaluate the CPs in the context of a humanoid robot. Typical reaching movements were collected with a Sarcos Sensuit, a device to record joint angular movement from human subjects, and approximated and reproduced with our imitation techniques. Our results demonstrate (a) that multi-joint human movements can be encoded successfully by the CPs, (b) that a learned movement policy can readily be reused to produce robust trajectories towards different targets, (c) that a policy fitted for one particular target provides a good predictor of human reaching movements towards neighboring targets, and (d) that the parameter space which encodes a policy is suitable for measuring to which extent two trajectories are qualitatively similar.

1 Introduction

This article addresses the question of how to encode desired trajectories to be performed by a humanoid robot. This question is particularly relevant to learning by imitation. Let us imagine, for instance, a situation in which some arm movements are demonstrated to the robot. Let us also assume the robot is able to extract the relevant kinematics of the movements, such as the joint-angles of the different degrees of freedom (DOFs), for instance. Finally, let us assume to robot is provided with inverse dynamics algorithms for reproducing the movement. The problems which remain to be solved are (1) how to store the demonstrated movements for future reuse, and (2) how to modify them for new external conditions, e.g for faster execution or for reaching another end position.

In [1], we identified five desirable properties that such a trajectory formation system should possess, namely: (1) the ease of representing and learning a goal trajectory, (2) compactness of the representation, (3) robustness against perturbations and changes in a dynamic environment, (4) ease of re-use for related tasks and easy modification for new tasks, and (5) ease of categorization for movement recognition. To the best of our knowledge, no system has yet been developed which combines all five properties. Previous approaches range from memorizing the entire trajectory at the sampling rate of the control servo [2], using spline-based methods [3], using optimization criteria [4], or employing lookup tables and neural networks that represent global control policies [5]. These approaches do not satisfy all our desirable properties. For instance, memorized trajectories are easy to learn, but are hard to re-use for new tasks and not robust towards significant changes of the environment. Spline-based approaches have a more compact representation, but otherwise share most of the properties of memorized trajectories. Optimization approaches are computationally expensive and cannot replan rapidly when the environment changes, and neural network based control policies are very hard to learn for even moderately dimensional systems.

This article is part of our exploration of combining nonlinear dynamical systems with regression techniques for encoding trajectories. In [1], we developed a system based on a mixture of pattern generators built from simple nonlinear autonomous dynamical systems. An observed movement was approximated by finding a best fit of the mixture model to its data by a recursive least squares regression technique. The system was used in combination with a movement execution controller (an inverse dynamics controller) for replaying the trajectories in a humanoid simulation. It could robustly replay the trajectories by taking perturbations to the robot into account, and by modifying the desired trajectories accordingly. It could also replay the trajectories towards different goals than the one of the demonstrated, while keeping the same qualitative velocity profiles as the demonstrated movement. One disadvantage of that system was that each basis function (i.e. each pattern generator) was a system of differential equations, which means that the number of state variables could become large when a high-quality fit was required, i.e. when many basis functions were needed. In this article, we develop a new system of control policies (CPs) based on similar nonlinear dynamical systems, which has the same properties as the system in [1] but which uses much fewer state variables. The velocity profiles of the demonstrated movements are learned using gaussian basis

Proceedings of the IEEE-RAS International Conference on Humanoid Robots

functions and locally weighted regression. The advantage of this approach is that the number of state variables does not depend on the number of basis functions, and arbitrary accuracy can therefore be obtained with a fixed number of state variables.

We apply the CPs to a task involving the imitation of human movements by a humanoid robot. This experiment is part of a project in rehabilitation robotics —the Virtual Trainer project —which aims at using humanoid rigid body simulations and humanoid robots for supervising rehabilitation exercises in stroke-patients. This article presents how trajectories recorded from human subjects performing a reaching task towards a set of predefined targets can be reproduced using the CPs. In particular, the purpose of this reaching experiment is three-fold. First, we quantify how well these different trajectories can be fitted with our CPs. Second, we estimate to which extent a CP, which is fitted to a particular reaching movement, is a good predictor of the movements performed by the human subject towards neighboring targets. And third, we compare the parameters of the basis functions of the systems fitted to different trajectories in order to evaluate how similar the trajectories are in parameter space. This last point aims at demonstrating that the CPs are not only useful for encoding trajectories, but also for classifying them.

2 Dynamical Systems As Control Policies

For the purpose of learning how to imitate human-like movement, we choose to represent movement in kinematic coordinates, e.g., joint angles of a robot or Cartesian coordinates of an endeffector —indeed only kinematic variables are observable in imitation learning. Thus, CPs represent kinematic movement plans, i.e., a change of position as a function of state, and we assume that an appropriate controller exist to convert the kinematic policy into motor commands. A controller policy is defined by the following differential equations which specify the attractor landscape of the policy for a goal g:

$$\dot{v} = \alpha_v(\beta(g-y)-v) \quad (1)$$
$$\dot{y} = \alpha_y(p+v) \quad (2)$$
$$\dot{\mathbf{z}} = f(\mathbf{z}) \quad (3)$$

where

$$f(\mathbf{z}) = \begin{bmatrix} \alpha_r(g-r) \\ \alpha_s\left(-s+(1-s)\frac{b}{g^2}(g-r)^2\right) \end{bmatrix}, \ \mathbf{z} = \begin{bmatrix} r \\ s \end{bmatrix} \quad (4)$$

and $\alpha_v, \alpha_r, \alpha_s$, β and b are positive constants.

The attractor landscape of the policy can be adjusted by learning a nonlinear mapping p approximated by locally weighted regression [6]. A prediction $\hat{p}$ is given by the normalized weighted sum of all locally linear models $\hat{p}_i$:

$$\hat{p} = \frac{\sum_{i=1}^{N} \Psi_i \hat{p}_i}{\sum_{i=1}^{N} \Psi_i} \quad (5)$$

where

$$\hat{p}_i = w_i v \quad (6)$$

w_i denotes the parameter of each local model and v denotes an input. Ψ_i is a Gaussian basis function

$$\Psi_i = \exp\left(-\frac{1}{2\sigma_{s,i}^2}(s-c_{s,i})^2\right) \quad (7)$$

which determines the region of validity of each locally linear model. w_i for each basis function can be learned by incremental locally weighted regression (LWR) [6] on line assuming that the goal is known (see Section 3). With the choice of locally linear model, (2) can be written in the form

$$\dot{y} = \alpha_y\left(\frac{\sum_{i=1}^{N}\Psi_i w_i}{\sum_{i=1}^{N}\Psi_i} + 1\right) v. \quad (8)$$

Notice that when $w_i = 0$ for all i, (y, v) dynamics in (1) and (8) are reduced to a simplified linear second order system. For notational convenience, we assume that all state variables $v, y, \mathbf{z}$ are initially zero, and $g \neq 0$. Note that this dynamical system has a unique equilibrium point at $(v, y, \mathbf{z}) = (0, g, (g, 0)^T)$ assuming that $\left(\sum_{i=1}^{N}\Psi_i w_i / \sum_{i=1}^{N}\Psi_i\right)|_{s=0} + 1 > 0$.

Figure 1 demonstrates an exemplary time evolution of the equations. We identify the state y as the desired position that the policy outputs to the controller, and $\dot{y}$ as the desired velocity. v is an internal state whose dynamics guarantee the unique point attractor at g. $\mathbf{z}$ implements a timing signal that is used to localize the Gaussian (7). It is the internal state of this timing signal that allows generating much more complex policies than a policy that has no internal state, as is indicated by the movement direction reversal in Figure 1. As will be demonstrated later, the choice of representing a timing signal in form of autonomous differential equations will allow us to be robust towards external perturbations of the movement– essentially, we can slow down, halt, or even reverse "time" by means of coupling terms in the $\mathbf{z}$ dynamics (see below).

2.1 Properties of the Policy Representation

2.1.1 Stability

Stability can be analyzed separately for the $\mathbf{z}$ dynamics and the (y, v) dynamics. To see that $\mathbf{z}$ is globally asymptotically stable, we shift the equilibrium point of the dynamics (3) to the origin using the change of variables $\tilde{r} = r - g$:

$$\begin{aligned} \dot{\tilde{r}} &= -\alpha_r \tilde{r} \\ \dot{s} &= \alpha_s\left(-s+(1-s)\tfrac{b}{g^2}\tilde{r}^2\right) \end{aligned} \quad (9)$$

Consider a Lyapunov function candidate

$$V(\tilde{\mathbf{z}}) = \frac{1}{2\alpha_r}\tilde{r}^2 + \frac{2g^4}{\alpha_s b^2}s^2 + \frac{1}{4\alpha_r}\tilde{r}^4 \quad (10)$$

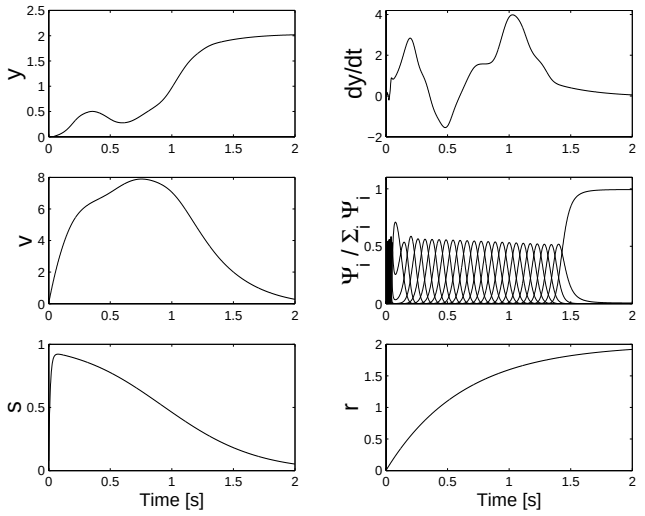

Figure 1: Time evolution of the dynamical systems. The number of RBFs is N=25, α_v=4 [1/s], α_y=0.2 [1/s], α_r=1.6 [1/s], α_s=5 [1/s], β=5, and b=15. The $c_{s,i}$ are spread out between $c_{s,1}$=0.2 and $c_{s,N}$=0.92 in order to be equally spaced in time. The same parameters will be used throughout the article. In this particular example, g=2 and the parameters of the basis functions were set by hand to arbitrary values.

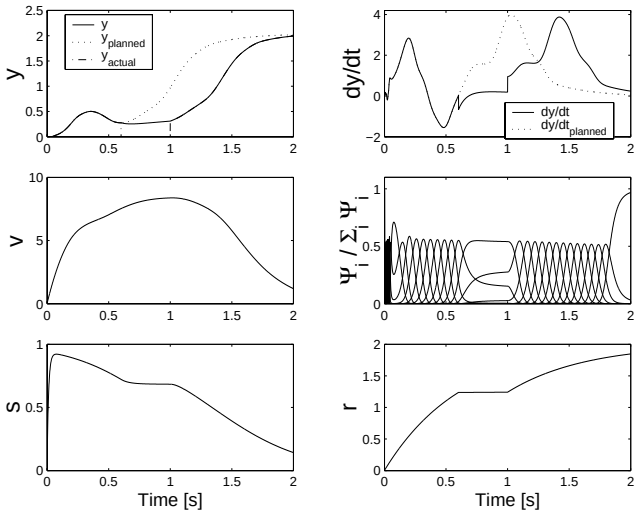

Figure 2: Time evolution of the dynamical systems under a perturbation (actual position $\tilde{y}$ set to zero between 0.6 and 1.0s). For this example $\alpha_{py} = 10$ and $\alpha_{pr} = 1000$ are used in (14) and (15) respectively.

where $\tilde{\mathbf{z}} = [\ \tilde{r},\ s\ ]^T$. Then, $\dot{V}(\tilde{\mathbf{z}})$ is given by

$$\dot{V}(\tilde{\mathbf{z}}) = -\tilde{r}^2 - \frac{4g^2}{b}\tilde{r}^2 s^2 - \left(\tilde{r}^2 - \frac{2g^2}{b}s\right)^2. \qquad (11)$$

$V(\tilde{\mathbf{z}})$ is positive definite for all $\tilde{\mathbf{z}} \in R^2$, radially unbounded, and $\dot{V}(\tilde{\mathbf{z}})$ is negative definite for all $\tilde{\mathbf{z}} \in R^2$. Thus, $\tilde{\mathbf{z}} = 0$ is globally asymptotically stable, which implies that $\mathbf{z} = (g, 0)$ is globally asymptotically stable.

We now investigate long time behavior of the rest of the state variables. For $t \to \infty$, we have

$$\dot{v} = \alpha_v(\beta(g - y) - v) \qquad (12)$$
$$\dot{y} = \alpha_\infty v \qquad (13)$$

where

$$\alpha_\infty = \alpha_y \lim_{s\to 0}\left(\frac{\sum_{i=1}^N \Psi_i w_i}{\sum_{i=1}^N \Psi_i} + 1\right) = const$$

since $s \to 0$ as $t \to \infty$. It is straightforward to see that $y \to g$ and $v \to 0$ as $t \to \infty$ as long as $\alpha_\infty > 0$.

2.1.2 Spatial and Temporal Scale Invariance

Spatial invariance of (3) requires that a scaling of the goal g does not affect the topology of the attractor landscape of the policy. Summarizing all state variables as $\mathbf{q} = [\ v,\ y,\ \mathbf{z}^T\ ]^T$ and abreviating $\dot{\mathbf{q}} = F(\mathbf{q})$, such topological equivalence [7] can be proven by finding a homeomorphism H that fulfills $H(\dot{\mathbf{q}}) = F(H(\mathbf{q}))$. For scaling the g by a factor c, $g \to cg$, it is easy to show that $H(\mathbf{q}) = [\ cv,\ cy,\ cr,\ s\ ]^T$ is an adequate homeomorphism.

Temporal scaling can be accomplished by scaling all the time parameters of the differential equations uniformly, α_v, α_y, α_r and α_s.

2.2 Robustness against Perturbations during Movement Execution

In the current form, the CP would create a desired movement irrespective of the ability of the controller to track the desired movement. When considering applications of our approach to physical systems, e.g., robots and humanoids, interactions with the environment may require an on-line modification of the policy. As an example, we show how a simple feedback coupling term can implement a stop in the time evolution of the policy.

Let $\tilde{y}$ denote the actual position of the movement system. We introduce the error between the planned trajectory y and $\tilde{y}$ to the differential equations (2) and (3) in order to modify the planned trajectory according to external perturbations:

$$\dot{y} = \alpha_y(\hat{p} + v) + \alpha_{py}(\tilde{y} - y) \qquad (14)$$
$$\dot{r} = \alpha_r(g - r)\left(1 + \alpha_{pr}(\tilde{y} - y)^2\right)^{-1} \qquad (15)$$

Figure 2 illustrates the effect of a pertubation where the actual position is artificially fixed during a short period of time. During the perturbation, the time evolution of the states of the policy is gradually halted. As soon as the perturbation stops, the system rapidly returns to the planned trajectory. Such on-line modifications are one of the most interesting properties of using autonomous differential equations for control policies.

3 Learning From Imitation And Classification

Assume we extracted a desired trajectory y_{demo} from the demonstration of a teacher. Adjusting the CP to embed

this trajectory in the attractor landscape is a linear function approximation problem. The demonstrated trajectory is shifted to a zero start position, and the time constants of the dynamical system are scaled such that the time dynamics $\mathbf{z}$ reaches zero at the same time as the goal state is reached in the target trajectory. Given that the goal state is known, it is guaranteed that the policy will converge to the goal; only the time course to the goals needs to be adjusted.

At every moment of time, we have v as input to the learning system, and as target output we have $target = \frac{\dot{y}_{demo}}{\alpha_y} - v$. We use recursive least squares (RLS) with a forgetting factor of λ [6] to determine the parameters w_i on line assuming that the goal is known.

Given a training point $(v, target)$, w_i is updated by

$$w_i^{t+1} = w_i^t + P_i^{t+1} v \dot{e} \tag{16}$$

where

$$P_i^{t+1} = \frac{1}{\lambda}\left(P_i^t - \frac{(P_i^t v)^2}{\frac{\lambda}{w_i} + v^2 P_i^t}\right), \dot{e} = target - w_i v$$

Despite this learning problem being a nonstationary learning task due to the effect of the change of the parameters w_i on the input distribution throught the variable v, very rapid learning is accomplished, and a single learning run through the target trajectory is usually sufficient for the learning to converge[1]. In practice, we set $w_1 = 0$ in the course of learning so that we expect that $\alpha_\infty \simeq \alpha_y$.

4 Experimental evaluations

4.1 Humanoid robot implementation

One of the motivations to develop the CPs comes from our Virtual Trainer (VT) project. The aim of this project is to supervise rehabilitation exercises in stroke-patients by (a) computerized learning the exercises from demonstrations by a professional therapist, (b), demonstrating the exercise to the patient with a humanoid simulation or robot, (c) video-based monitoring the patient when performing the exercise, and (d) evaluating the patient's performance, and suggesting and demonstrating corrections with the simulation when necessary.

In this article, we tested the CPs for a learning by imitation task in a humanoid robot with 30 DOFs [9]. The robot is a 2-meter high hydraulic anthropomorphic robot with legs, arms, a jointed torse, and a head. The robot is fixed at the hip, which means that it does not require balance. The CPs are used in combination with a motor system which uses inverse dynamics for generating the torques necessary to accurately reproduce a desired trajectory. As will be described next, the task involves arm reaching movements towards specific points on a vertical plane. As such, it is not part of a standard rehabilitation exercize, but rather represents an experiment for quantitatively evaluating the CPs.

[1]Note that the locally linear models we use have singurality at $v = 0$. In practice, this singularity during fitting only occurs with a few trajectories which we have recorded. A solution to this problem is provided in [8].

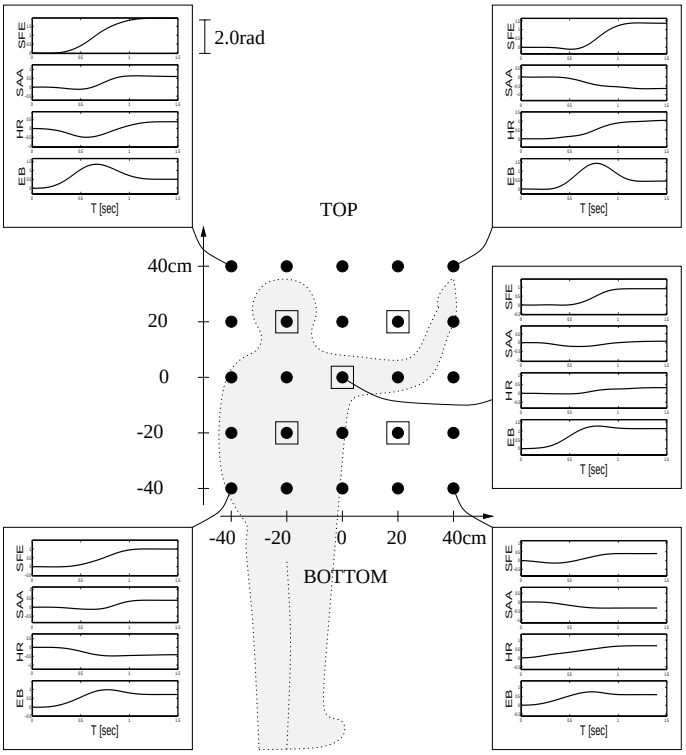

Figure 3: Grid containing the 25 reaching targets with 5 typical reaching trajectories. Each trajectory is represented by the joint angles of four degrees of freedom drawn from top to bottom: shoulder flexion-extension (SFE), shoulder adduction-abduction (SAA), humerus rotation (HR), and elbow flexion-extension (EB). The five points surrounded by a square were used as test sets for the experiment described in Section 4.2.3.

4.2 Results

4.2.1 Recording of trajectories from human subjects

We recorded trajectories performed by a human subject using a joint-angle recording system, the Sarcos Sensuit (Figure 8 left). The Sensuit directly records the joint angles of 35 DOFS of the human body at 100Hz using hall effect sensors with 12 bit A/D conversion.

We recorded reaching trajectories towards 25 points located on a vertical plane placed 65cm in front of the subject. The points were spaced by 20cm on a 80 by 80cm grid (Figure 3). The middle point was centered on the shoulder, i.e it corresponded to the (perpendicular) projection of the shoulder on the plane. The reaching trajectories to each target were repeated three times.

Figure 3 shows five typical reaching trajectories in joint space. As could logically be expected, the trajectories present qualitative differences between movements to the left and right, with the shoulder adduction-abduction (SAA) angle increasing and decreasing respectively (adduction corresponds to a positive SAA angle). Another qualitative difference is observable between humerus rotation (HR) towards bottom and top targets on the left side. The differences in shoulder flexion-extension (SFE) angles

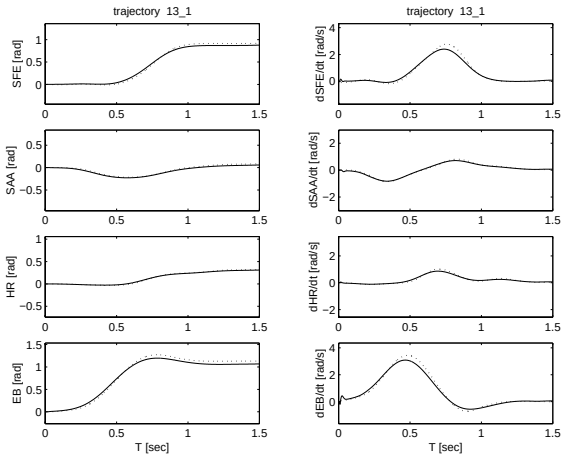

Figure 4: Fitting of a reaching movement with 4 DOFs shoulder flexion-extension (SFE), shoulder adduction-abduction (SAA), humerus rotation (HR) and elbow rotation (EB) towards a target located at <0,0>cm. The recorded and fitted angles of the DOFs are drawn in dotted and continuous lines, respectively.

and elbow flexion-extension (EB) are essentially quantitative rather than qualitative. The distances between initial and final SFE angles are, for instance, approximately twice as large for the reaching towards high targets compared to reaching towards low targets.

4.2.2 Fitting each reaching movement

A reaching movement is fitted by the CPs as four independent trajectories, one for each degree of freedom (SFE, SAA, HR and EB). Each of these trajectory is fitted by a system of 25 basis functions. One reaching movement is therefore encoded by a total of 100 parameters.

We fitted the total of 75 reaching movements performed. Figure 4 compares the recorded and fitted trajectories for one particular example. The figure illustrates that 25 basis functions per degree of freedom are sufficient to fit the demonstrated movements with little residual error.

4.2.3 Modulation of the goal of the reaching movement

We tested how well the CP fitted to one particular trajectory could be used as a predictor of the trajectories performed by the human subject towards neighboring targets. Five control policies —those fitted to the five targets surrounded by a a square in Figure 3— were tested.

The test is performed as follows. Once a CP is fitted to one particular reaching movement, its parameters are fixed and only the goals g, i.e. the four angles (one per DOF) to which the CP will converge, are modulated. The different goals that we feed into the CP are then provided by the final postures of the human subject when reaching towards different target points. This therefore allows us to directly compare the angular trajectories predicted by the CPs with those performed by the subject.

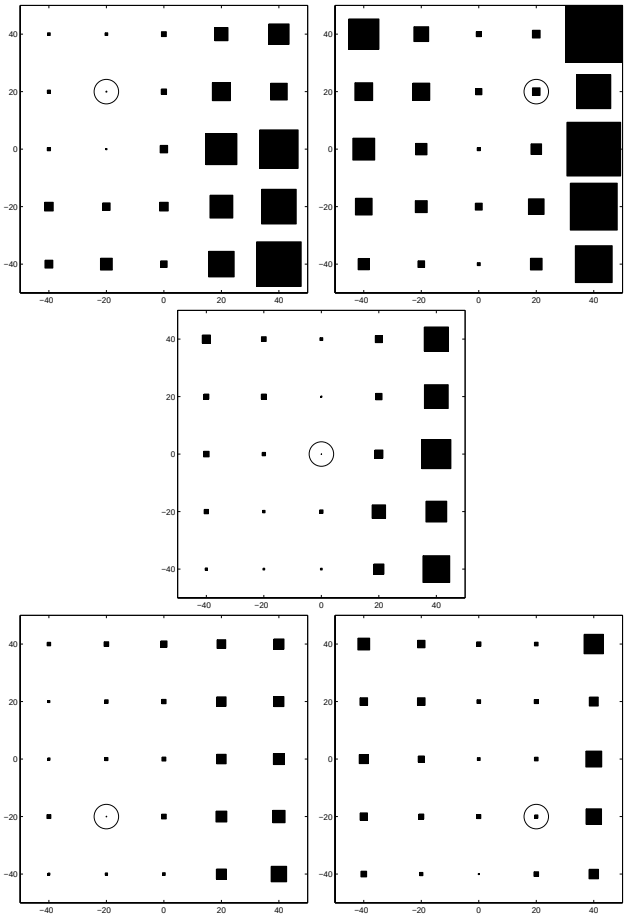

Figure 5: Error between predictions of the TFSs and recorded movements. The targets of the recorded movements which were used for fitting each of the five TFS are shown with circles. The widths of the squares are proportional to the square error measurements as described in the text.

The accuracy of the prediction is measured by computing the square error between the recorded and predicted velocities normalized by the difference between the maximum and minimum velocities of the recorded movement $\sum_i (\frac{v_i^{\text{rec}} - v_i^{\text{pred}}}{max(v_i^{\text{rec}}) - min(v_i^{\text{rec}})})^2$, summed for the four degrees of freedom, and averaged over the three instantiations of each recorded movement.[2]

Figure 5 shows the resulting square error measurements between predictions and recorded movements for the five chosen CPs and Figure 6 illustrates a predicted trajectory by one of the CP. Two observations can be made. First, a CP tends to be a better predictor of recorded movements which were made towards a target close to the target for which it was fitted. This is logical, as it simply means that human reaching movements towards close by goals tend to follow similar trajectories. Note that, because of

[2]Note that this error measure is relatively crude and tends to overestimate how different two trajectories are. It is sensitive to differences between the actual starts of the movements. Two trajectories with very similar profiles can potentially have a large error if they do not start at the same time.

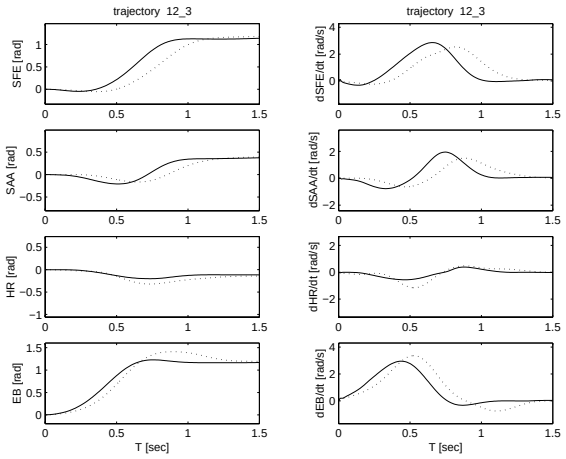

Figure 6: Comparison of a recorded and predicted trajectory. Dotted lines show a recorded trajectory towards a target located at <-20,0>cm. Continuous lines show the prediction from a CP which has been fitted with a trajectory towards a target located at <-20,-20>cm, and which has been given new angle goals.

the variability of human movements, the prediction error for the recorded trajectories towards the same targets as those used for the fitting are not zero, as the two other instantiations might vary slightly from the trajectory used for the fitting. Second, a CP is a significantly better predictor of recorded movements which are made towards targets located above or below the original target, compared to targets located left or right to that target. This is in agreement with our observation that human trajectories in our recordings present qualitative differences between left and right —which therefore require different CPs—, and only quantitative differences between top and bottom —which can be represented by a single CP with different goals.

Note that the purpose of our CPs is not to explain human motor control, and there is no reason to expect that a single CP could predict all these human reaching movements. As mentioned above, the movements exhibit qualitative differences in joint space. One would need some cartesian space criterion (e.g straight line of the end point of the arm) and some optimization criterion (e.g. minimum torque) to explain human motor control. Our purpose is rather to develop a system to be used in humanoid robotics which can accurately encode specific trajectories, replay them robustly, and modulate them to different goals easily. Here we observe that, for the specific trajectories that we recorded from human subjects, a CP fitted towards a particular target has the additional property of being a a good predictor for human movements to neighbor targets.

4.2.4 Comparison of trajectories in parameter space

In the last test, we compared the parameters w_i fitted to the different reaching movements. As a measure of similarity, we computed the average correlation between set

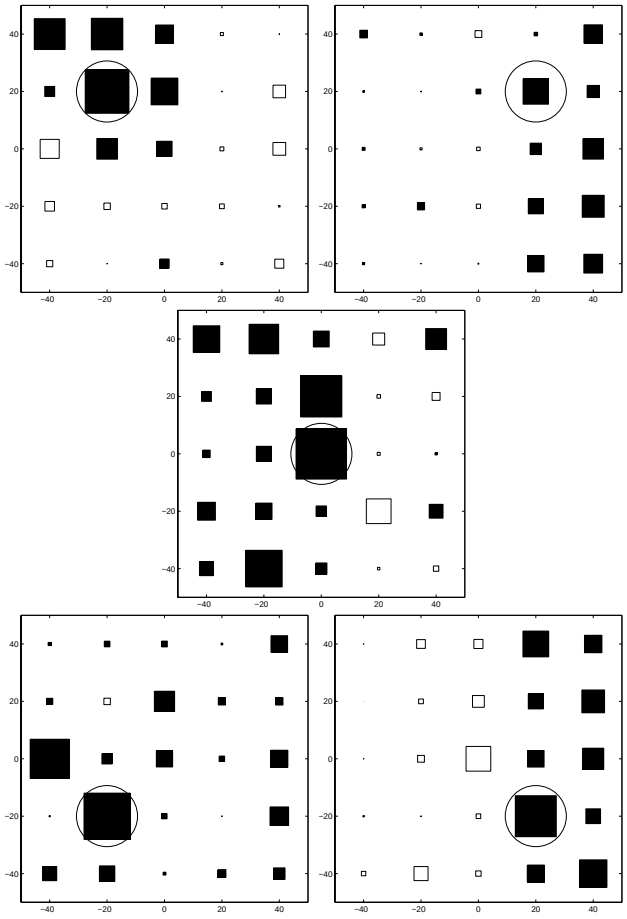

Figure 7: Averaged correlation between parameters w_i of different CPs. The square width is proportional to the absolute value of the correlation as described in the text. Squares filled with black and white indicate positive and negative correlation, respectively. The circles indicate the target movements with which all the others are compared.

of parameters, as follows: $\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{1}{9}\frac{\mathbf{w}_a^i\mathbf{w}_b^j}{|\mathbf{w}_a^i||\mathbf{w}_b^j|}$ where $\mathbf{w}_a^i$ is the vector containing all the parameters of the CPs fitted with the ith (out of three) reaching movement towards target a.

Figure 7 shows the correlations for five chosen CPs. A general trend is that parameters w_i of CPs fitted to trajectories towards the same targets tend to have high positive correlation and that the correlation decreases (sometimes changing sign) for trajectories towards distant targets. High correlation of parameters can therefore be used as a measure of the similarity between trajectories. Related to our observation that trajectories towards different heights present only quantitative differences, the correlations tend to be higher along vertical axes rather than horizontal axes. Qualitatively similar movements tend therefore to have similar parameters.

In [8], we demonstrated how a very similar system could be used for classification of movements made for the drawing of 2D graffiti alphabet characters. For that experiment, trajectories for a particular character could be rec-

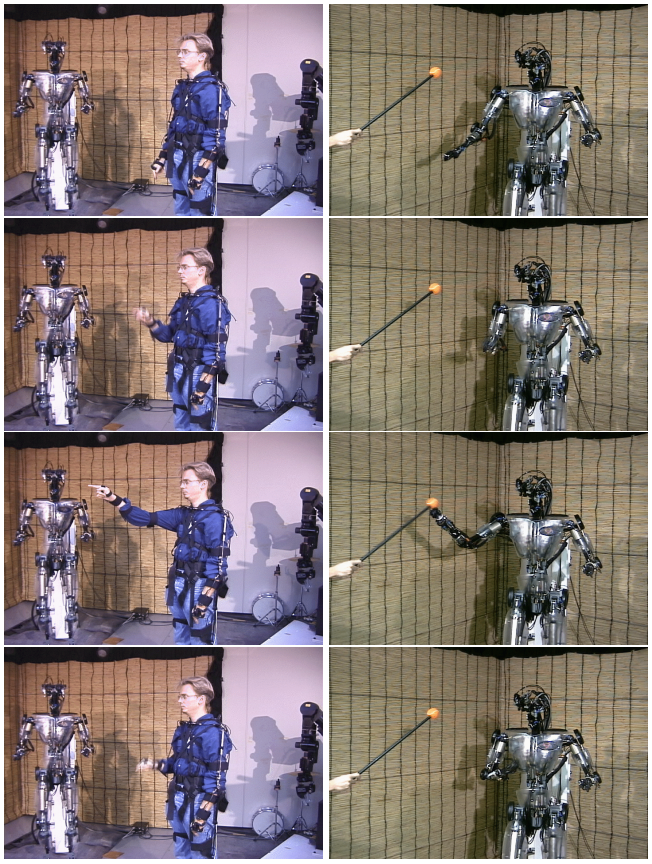

Figure 8: Humanoid robot reproducing reaching movements following a human demonstration.

ognized by simply using a highest correlation criterion of their weights of the RBF network. In the present experiment, the different trajectories are less distinct and more part of a continuum.

4.2.5 Robot implementation

Figure 8 shows the humanoid robot performing a reaching movement. The CPs fitted in the previous section were used to perform the movements. Two sets of CPs were used for one complete reaching movement: the reaching trajectory as shown in the previous sections, and the return trajectory which brings the arm back to the resting position (along the body). In our experiment, we varied the cartesian targets of the reaching trajectory, such as to investigate how realistic the movements appeared towards different targets. Using a color-tracking system with two cameras, the position of the target (an orange ball) was computed, and fed into an inverse kinematic algorithm to compute end goals in joint-angle space. New trajectories were then generated by the CPs and fed into the inverse dynamics system for motor execution. In these experiments, we did not perturb the arm movements of the robot, which means that the trajectories followed by the arm are very similar to those reported in the previous figures. Similarly to the previous sections, we observed that the reaching movements made to targets which are located above or below the original targets looked very similar to the demonstrated movement. Overall, the reaching movements looked strikingly human-like.

5 Discussion

This article is part of our exploration of combining nonlinear dynamical systems with regression techniques for encoding trajectories, and follows from preliminary experiments presented in [1]. Two key characteristics of the CPs are 1) that the trajectory is not indexed by time but rather develops out of integrating autonomous nonlinear differential equations, and 2) that the CP does not encode one single specific desired trajectory but rather a whole attractor landscape with a unique point attractor, the goal state. It should be noted that our suggested approach to trajectory formation is in strong contrast with methods which explicitly index the trajectory by time, such as spline fitting between via points. Explicit time indexing makes is hard to modify a trajectory in response to dynamically changing environments and strong perturbations that are beyond the abilities of a PD servo. Moreover, time-indexed trajectories are specific to initial conditions and re-using of the trajectory for new movement goals is complicated.

Our approach is inspired by the concepts of pattern generators [10] and force fields [11] found in biology. Related work includes [12, 13, 14]. The CPs presented here are most closely related to the VITE model [12], but with the extension that we use multiple basis functions (the Ψ_i functions) to allow generating a larger class of movements, and that our dynamical systems do not require artificial resetting of the states of the dynamical systems after a movement. The concept of combining multiple basis functions has parallels in statistical learning theory [15, 16].

Our CPs are interesting candidates for satisfying the desirata enumerated in the introduction:

1. *Ease of representing and learning a goal trajectory.* The CP can acquire desired trajectories by on-line recursive least squares regression techniques assuming that the goal is known. This property compares favorably in terms of learning speed and convergence properties towards other approaches of trajectory learning. Our only requirement is that trajectories end with zero-velocity, i.e., that they represent discrete movement. In our work, a set of 25 basis functions was sufficient to represent trajectories lasting between 1.0 and 3.0 seconds. More basis functions can be added for longer sequences, and/or for representing finer details of movement.

2. *Compactness.* The encoding in a mixture of dynamical systems is comparable in compactness with spline fitting using via points as movements are encoded by few parameters, namely the N parameters w_i and the goals G for each trajectory.

3. *Robustness against perturbations and dynamically changing environments.* As illustrated in this article, the ability to smoothly recover from perturbations is a primary feature of the dynamical systems we used. Furthermore, the CPs give a good basis for dealing with perturbations at a planning level instead of the execution level. For instance, recovering from a perturbation requires dif-

ferent actions depending on the purpose of the imitation task. If the purpose is to respect the subgoals of a movement despite the perturbation, the timing of the subgoals can be modified such that the next subgoal is fed into the system only when the current goal is reached within a satisfactory limit. On the other hand, if the purpose is to respect the timing at all costs, subgoals can be fed in at their original timings, and the CPs will naturally modify the desired trajectory during the perturbation, and skip desired subgoals (without building up a potentially huge torques in the robot's motors, as could happen in a PD controller).

4. *Ease of modification.* Desired trajectories can readily be modified by manipulating the values and timings of the subgoals, and by adding additional coupling terms. For instance, if the aim of a movement is to reach a particular object with a particular velocity profile (e.g. a tennis serve), the CP can learn that particular velocity profile and reuse it in multiple occasions by adapting the goal to the current location of the object (if necessary, using inverse kinematics for finding the desired end positions of each DOF). Another possibility is to use potential field approaches to add attracting or repelling terms into the differential equations to navigate obstacles. In this article, we have demonstrated that CPs which have learned a trajectory to one specific target, can readily generate trajectories to different targets, and that these newly generated trajectories present many similarities to those of the human subject when the targets are close to the original target.

5. *Ease of categorization.* An important question in learning movements by imitation is how to recognize similar movements and which metrics to use to measure differences between movements. In this article, we showed that computing the correlation of parameters of two CPs is a good first approach to get an estimation of how qualitatively similar two trajectories are. In a related article [8], we have demonstrated that such a simple measure is sufficient to distinguish different 2D characters drawn with a mouse. In future work, we intend to investigate in more detail different metrics for comparing movements in the CPs' parameter space.

Acknowledgements

This work was made possible by support from the US National Science Foundation (Awards 9710312 and 0082995), the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Cooperation, and the ATR Human Information Processing Research Laboratories.

References

[1] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*. 2001. In press.

[2] C.H. An, C.G. Atkeson, and J.M. Hollerbach. *Model-based control of a robot manipulator*. MIT Press, 1988.

[3] H. Miyamoto, S. Schaal, F. Gandolfo, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. A kendama learning robot based on bi-directional theory. *Neural Networks*, 9:1281–1302, 1996.

[4] M. Kawato. Trajectory formation in arm movements: minimization principles and procedures. In H.N. Zelaznik, editor, *Advances in Motor Learning and Control*, pages 225–259. Human Kinetics Publisher, Champaign Illinois, 1996.

[5] R. Sutton and A.G. Barto. *Reinforcement learning: an introduction*. MIT Press, 1998.

[6] S. Schaal and C.G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084, 1998.

[7] J. Guckenheimer and P. Holmes. *Nonlinear Oscillators, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, 1983.

[8] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE International Conference on Robotics and Automation*. 2002. Submitted.

[9] C. G. Atkeson, J. Hale, M. Kawato, S. Kotosaka, F. Pollick, M. Riley, S. Schaal, S. Shibata, G. Tevatia, and A. Ude. Using humanoid robots to study human behaviour. *IEEE Intelligent Systems*, 15:46–56, 2000.

[10] F. Delcomyn. Neural basis for rhythmic behaviour in animals. *Science*, 210:492–498, 1980.

[11] S.F. Giszter, F.A. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13:467–491, 1993.

[12] D. Bullock and S. Grossberg. VITE and FLETE: neural modules for trajectory formation and postural control. In W.A. Hersberger, editor, *Volitional control*, pages 253–297. Elsevier Science Publishers, 1989.

[13] D. Kleinfeld and H. Sompolinsky. Associative network models for central pattern generators. In C. Koch and I. Segev, editors, *Methods in neural modeling*, pages 195–246. MIT Press, 1989.

[14] F.A. Mussa-Ivaldi. Nonlinear force fields: a distributed system of control primitives for representing and learning movements. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 84–90. IEEE, Computer Society, Los Alamitos, 1997.

[15] G. J. McLachlan and K. E. Basford. *Mixture Models*. Marcel Dekker, 1988.

[16] M.I. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.