Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001

Trajectory Formation for Imitation with Nonlinear Dynamical Systems

Auke Jan Ijspeert[1,2], Jun Nakanishi[1] and 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@usc.edu, sschaal@usc.edu

Abstract

This article explores a new approach to learning by imitation and trajectory formation by representing movements as mixtures of nonlinear differential equations with well-defined attractor dynamics. An observed movement is approximated by finding a best fit of the mixture model to its data by a recursive least squares regression technique. In contrast to non-autonomous movement representations like splines, the resultant movement plan remains an autonomous set of nonlinear differential equations that forms a control policy which is robust to strong external perturbations and that can be modified by additional perceptual variables. This movement policy remains the same for a given target, regardless of the initial conditions, and can easily be re-used for new targets.

We evaluate the trajectory formation system (TFS) in the context of a humanoid robot simulation that is part of the Virtual Trainer (VT) project, which aims at supervising rehabilitation exercises in stroke-patients. A typical rehabilitation exercise was 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 that multi-joint human movements can be encoded successfully, and that this system allows robust modifications of the movement policy through external variables.

1 Introduction

An important issue in humanoid robotics, and in learning by imitation in particular, is the question of how to encode desired trajectories to be performed by the robot. Various approaches have been suggested in the literature, ranging from memorizing the entire trajectory at the sampling rate of the control servo [1], using spline-based methods [2], using optimization criteria [3], or employing lookup tables and neural networks that represent global control policies [4]. In general, there seems to be consensus that among the most important desirable properties of movement encoding are: 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. When examining standard approaches of movement planning and representation against this check list, it becomes obvious that no approach exists that accomplishes all these goals. 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 re-plan rapidly when the environment changes, and neural network based control policies are very hard to learn for even moderately dimensional systems.

In this article, we explore an alternative to these approaches which can fulfill all the desired characteristics above. The idea is to encode desired trajectories, or more precisely complete control policies, in terms of a mixture of pattern generators built from simple nonlinear autonomous dynamical systems. In contrast to other approaches in the literature, the dynamical systems encode *desired* trajectories, not motor commands, such that an additional movement execution stage by means of a standard controller (e.g., inverse dynamics controller in our work) is needed. This strategy, however, does not prevent us from modifying the trajectory plans on-line through external variables, as will be demonstrated later.

Our current work is restricted to discrete movements, i.e., movement with a unique point attractor. Complex movement can be represented by a sequence of such mixture model movement primitives. The transition from one movement segment to another could be state-triggered or time-indexed, depending on the task to be accomplished – this is similar to via-point approaches suggested by others [2]. Given that the component dynamical systems are globally stable, our suggested weighted superposition will be globally stable, too.

We apply this trajectory formation system (TFS) to a task involving the imitation of human movements by a humanoid simulation. 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 of typical rehabilitation exercises recorded from human subjects can be reproduced by the simulator using the trajectory formation system, and that this kind of encoding

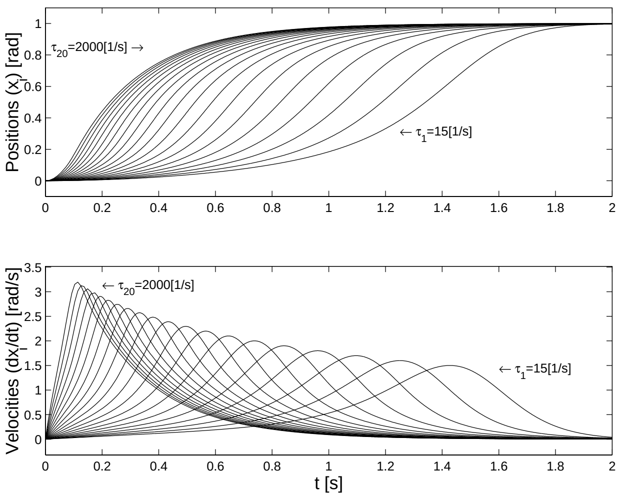

Figure 1: Set of primitives for one DOF. Top: trajectories, Bottom: velocity profiles. The following parameters are used throughout the paper: $N = 20$, $\gamma = \varepsilon = 0.005$, $\beta = 4.0$[1/s], $\mu = 3.0$[1/s], and $\tau_i = [15, 19.4, 25.1, \ldots, 2000]$[1/s]. G was here set to 1.0 and no perturbation was fed back into the system, i.e. $\theta_R = \theta$.

fulfills all the desired characteristics enumerated above.

In the next sections, we first present our trajectory formation system and its extensions to imitation in Section 2. Then we briefly describe the Virtual Trainer project and the requirements it places on the trajectory formation system (Section 3.1), our humanoid simulation (Section 3.2), and the method for recording human movements (Section 3.3). Results of multi-joint fitting are finally presented in Section 3.4. Our approach is discussed as well as compared with alternative methods in Section 4.

2 Dynamical systems for trajectory formation

In earlier work, we have developed nonlinear dynamical systems for discrete and rhythmic movements in a humanoid robot[5, 6]. Trajectories for each degree of freedom were created by superimposing two dynamical systems, one for discrete (point-attracting) movement, the other for rhythmic (limit-cycle) movement. The system could produce robust trajectory planning (e.g. perturbations did not prevent from reaching specific targets, and did not disrupt smooth and stable movement execution) by numerically integrating the differential equations of the dynamical systems, and was successfully used to control juggling and drumming tasks. The applicability of this system, however, was limited to simple reaching movement as only monotonic movement could be realized with symmetric bell-shaped velocity profiles. For instance, creating a tennis fore-hand swing would not be feasible with this encoding. In this paper, we generalize movement planning with dynamical systems by developing a trajectory formation system based on mixtures of nonlinear dynamical systems, which allow accurate reproduction of almost arbitrary human movements.

Our approach presented in this paper is inspired by the idea of basis function and mixture models in statistical learning [7]. Instead of representing a function with one complicated representation, it is often easier to accomplish the same goal with a representation of superimposed simple component systems. Spline-based trajectory encodings are essentially such mixture systems, except that they are non-autonomous and do not form a stable control policy. Thus, instead of splines, we would like to use autonomous dynamical system as components with well-defined point attractors. The weighted superposition of these component system, or movement primitives, can form very complex attractor landscapes on the way to the equilibrium point. Complex movements that have intermediate starts and stops need to be segmented at these "via points" such that every segment is fitted separately by a mixture of movement primitives.

As a first approach, we created the following family of N primitives for encoding the trajectory of one DOF:

$$\dot{v_i} = \tau_i \cdot \left(\gamma + \frac{x_i^2}{G^2+\varepsilon}\right) \cdot (-v_i + \beta(G - x_i) + \mu(\theta_R - \theta)) \quad (1)$$

$$\dot{x_i} = v_i \quad (2)$$

$$\theta = \sum_i^N w_i x_i \quad (3)$$

$$\dot{\theta} = \sum_i^N w_i v_i \qquad \text{with} \qquad \sum_i^N w_i = 1 \quad (4)$$

where θ and $\dot{\theta}$ are the desired angular position and velocity to be performed by the DOF, and G is the desired displacement (or goal position). θ_R is the actual joint angle realized by the DOF. The state variables x_i and v_i are the positions and velocities of each individual primitive i. Primitives are determined by four strictly positive parameters —two time constants τ_i and β, and a parameter γ which determines the width of the velocity bell shape— one positive gain parameter μ, and a small constant ε. The contribution of each primitive to the movement is determined by the weights w_i.

By construction, the position x_i of each primitive converges to the goal with a bell-shaped velocity profile (Figure 1). Note that, as the weights w_i are not restricted to be positive, the $\theta(t)$ trajectories do not need to be monotonic. Perturbations to the system which prevent the real angle θ_R to be equal to the desired angle θ (e.g. due to external forces applied to the DOF) are fed back into the system by the $\mu(\theta_R - \theta)$ term in equation 2, effectively slowing down or terminating the planning dynamics of the mixture model.

In our experiments, the trajectory formation system (TFS) will be based on a set of $N = 20$ primitives per DOF in which the parameters β, γ and ε are fixed and identical for all primitives, while τ_i varies according to $\tau_i = \tau_{min}(\tau_{max}/\tau_{min})^{\frac{i-1}{N-1}}$ to ensure more or less equal spacing between the maxima of the velocities (Figure 1) .

Despite its simplicity, the mixture of primitives possesses a variety of appealing properties:

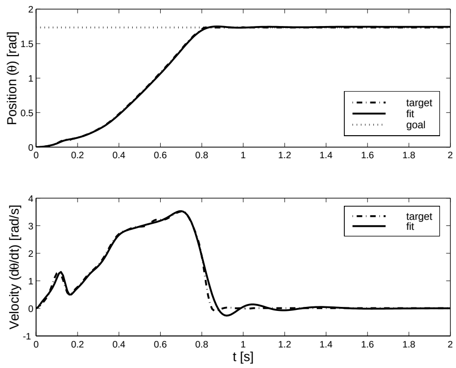

Figure 2: Example of fitting a desired trajectory using recursive least squares regression. Top: position trajectory, Bottom: velocity trajectory.

1. the velocity profiles of each individual primitive is smooth and has a strong resemblance to the bell-shaped profiles observed in human movements – this will be beneficial for smooth trajectory generation,
2. the mixture of primitives can be fitted to a desired trajectory using an incremental least squares regression on line assuming that the goal is known (see below),
3. individual primitives, as well as linear combinations of them, are stable systems which all converge to a unique point attractor corresponding to the goal when only transient perturbations are applied to them (see next section),
4. the complete system remains an autonomous dynamical system, i.e., there is no explicitly time dependency,
5. a particular movement can be characterized by the weight vector of the primitives—this has the potential to allow movement recognition based on these parameters.

2.1 Stability of the mixture model

This section presents a stability analysis of the proposed dynamical primitives for the case $\theta_R = \theta$.

By setting $\dot{x_i} = \dot{v_i} = 0$ and solving the equations for x_i and v_i, we find that each component of the mixture model has a unique equilibrium point at $(x_i, v_i) = (G, 0)$ [1]. To see that this equilibrium point is globally asymptotically stable, we shift it to the origin by the change of variables $\bar{x}_i = x_i - G$.

Consider a Lyapunov function candidate

$$V(\bar{x}_i, \bar{v}_i) = \frac{1}{2}\bar{v}_i^2 + \frac{\beta\tau_i}{12(G^2+\varepsilon)} \cdot \left(3\bar{x}_i^2 + 8G\bar{x}_i + 6\gamma\varepsilon + 6G^2(\gamma+1)\right)\bar{x}_i^2 \qquad (5)$$

Note that (5) is indeed positive definite for $\gamma, \varepsilon > 0$ since

$$\begin{aligned} & 3\bar{x}_i^2 + 8G\bar{x}_i + 6\gamma\varepsilon + 6G^2(\gamma+1) \\ = \; & 3(\bar{x}_i + \frac{4}{3}G)^2 + \frac{2}{3}G^2 + 6\gamma\varepsilon + 6G^2\gamma > 0. \end{aligned}$$

[1]The equilibrium is shifted to $\tilde{x}_i = G + \mu(\theta_R - \tilde{\theta})/\beta$ and $\tilde{v}_i = 0$ if a constant perturbation is applied to the DOF.

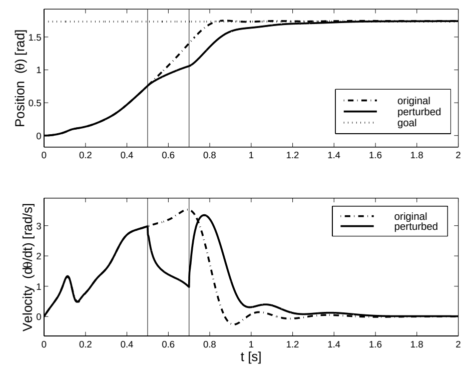

Figure 3: Recovery from a perturbation of the position (θ_R is forced to be 0.0 from t=0.5s to t=0.7s, while it is set to θ the rest of the time). Top: position trajectory, Bottom: velocity trajectory.

Then, $\dot{V}(\bar{x}_i, \bar{v}_i)$ is given by

$$\dot{V}(\bar{x}_i, \bar{v}_i) = -\tau_i\left(\gamma + \frac{(\bar{x}_i + G)^2}{G^2+\varepsilon}\right)\dot{\bar{v}}_i^2 \leq 0. \qquad (6)$$

Since

$$\dot{V}(\bar{x}_i, \bar{v}_i) = 0 \;\Rightarrow\; \bar{v}_i = 0 \;\Rightarrow\; \bar{x}_i = 0, \qquad (7)$$

the origin is asymptotically stable by applying LaSalle's invariance theorem. Moreover, since $V(\bar{x}_i, \bar{v}_i)$ is radially unbounded, the origin is globally asymptotically stable. Thus, the equilibrium point of each primitive dynamics, $(x_i, v_i) = (G, 0)$, is globally asymptotically stable. Furthermore, the mixture of these dynamical systems has a globally asymptotically stable attractor point at $(G, 0)$ as long as the sum of all the weights is 1 for the case $\theta_R = \theta$.

2.2 Learning from demonstration

The mixture of primitives learns a desired trajectory from the demonstration by adjusting the set of weights, w_i. We use recursive least squares [8] to adjust the weights on line assuming that the goal G is known[2].

Given a training point $(\dot{\mathbf{x}}, \dot{\theta}_{des})$, $\mathbf{w}$ is updated by

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \mathbf{P}^{t+1}\dot{\mathbf{x}}\dot{\mathbf{e}}^T \qquad (8)$$

where

$$\mathbf{P}^{t+1} = \frac{1}{\lambda}\left(\mathbf{P}^t - \frac{\mathbf{P}^t\dot{\mathbf{x}}\dot{\mathbf{x}}^T\mathbf{P}^t}{\lambda + \dot{\mathbf{x}}^T\mathbf{P}^t\dot{\mathbf{x}}}\right), \mathbf{e} = \theta_{des} - {\mathbf{w}^t}^T\mathbf{x}$$

and $\mathbf{x} = [\; x_1, \; \cdots, \; x_N \;]^T$, $\mathbf{w} = [\; w_1, \; \cdots, \; w_N \;]^T$. We choose to use velocity data for learning since we empirically find that this leads to smoother trajectories.

Figure 2 shows an example of the result of learning of the elbow joint angle during a reaching movement demonstrated by a human. The fit is based on the 20 primitives shown in Figure 1. These primitives were sufficient to provide a good fit of the trajectory, with very little residual error.

[2]λ is a forgetting factor which is normally set to 1.0 for stationary learning data (as is the case in this paper), but which might be set to a smaller value for non-stationary cases (e.g. when the goal G is not known in advance and has to be learned on line).