Robotics and Autonomous Systems 61 (2013) 340–350

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Guided latent space regression for human motion generation

Carlo Alberto Avizzano *

PERCRO Laboratory, The Institute for Telecommunication, Information Technologies and Perceptual Robotics (TeCIP), Scuola Superiore Sant'Anna, Pisa, Italy

ARTICLE INFO

Article history:
Available online 28 September 2012

Keywords:
Motion models
Trajectory generation
Programming by demonstration
Robust regression
Trajectory adaptation

ABSTRACT

In the present work, we describe a mathematical model to generate human-like motion trajectories in space. We use linear regression in a latent space to find the model parameters from a set of demonstration examples.

The learning procedure requires a relevant set of similar examples. The apprehended models encode both the typical shapes of motion and their variability towards specific boundary conditions (BC). We will show the added value of encoding both properties in a unique model and we apply this ability to common problems of error compensation and target tracking.

The models allow us to describe human motion using expansion-function series (EFS), thus avoiding typical stability issues that arise in the use of differential equation models. To cope with variable scenarios, we show two specific algorithms that morph and adapt the evolution trajectory. In analogy to splines, the EFS preserve an analytical structure on which we develop the optimisation steps. In such a way, we managed to combine multiple single segments into complex motions that preserve continuity and may simultaneously optimise other criteria.

In the present work, after having analysed similar tools, we present the basic model and its features. Then we develop a robust tool to gather the model from examples, and to achieve real-time trajectory adaptation. The achieved results will be analysed through an experimental analysis on data collected in a ball catching experiment.

1. Introduction

Many approaches exist so far to provide tools for modelling human and character motion through examples given by demonstration(s). These approaches mostly serve two types of applications: character animation in computer graphics and for motion planning in robots [1]. In the first case, the motions usually consider whole-body animation to moving visual avatars. In the second case, learning focuses on a limited set of body-coordinates, and uses the generated trajectories to provide inputs for motion control algorithms.

Motion programming, guided by demonstration examples, has several motivations. Programming by demonstration is intuitive, natural and easier even for non-skilled operators; it allows humans to exchange their knowledge in the same manner they do in everyday life. Demonstrations are direct and cost-effective. Traditional motion programming requires the humans to describe motion policies with respect to the environmental conditions. Conversely, programming by demonstrations automatically monitors the environmental conditions that cause actions. During a demonstration, the operator copes with several high-level problems and the learning procedure can catch the shown solution. In such a way the learned models embed several features, such as: complex dynamics, harmonious movements, knowledge of the environmental properties, suboptimal solutions (minimum torque or energy), matching of boundary conditions, and cinematic ranges.

According to the application, the design of motion models employs different requirements and performance criteria.

Character animation does not consider robust, physically based, and real-time operations as strict constraints. Simulations can be repeated several times to achieve satisfactory results. The most relevant goal is to achieve the most realism that is possible. Human examples provide information to estimate the joint torques applied during the examples. Using these torques, learning algorithms train articulated models similar to the human dynamics [2,3]. These models help to avoid unrealistic artefacts such as false equilibrium and moon-walking [4], and to implement complex environmental interactions such as manipulation [5], cooperation [6], or constrained motions [7].

In robot motion planning, the success of the interaction is of primary importance. The designed models control relevant real-time operation and should never fail. In this condition, other issues come to the designer's attention, such as: the model local and global stability [8], task refinement [9,10] and task generalisation [11].

Several types of models have been proposed to implement such functionalities. Most of them are represented as autonomous

* Tel.: +39 347 6320 199; fax: +39 050 883 333.
E-mail address: c.avizzano@sssup.it.
URL: http://www.percro.org.

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.09.001

Nomenclature

ES	Expansion Series
EFS	Expansion Function Series
DMP	Dynamic Motion Primitives
SEDS	Stable Estimator Dynamic Systems
M1	Primary Motor Cortex
PPC	Posterior Parietal Cortex
DST	Discrete Sine Transform
$q_{n,m}(t)$	Generalised motion coordinate n, of example mth
$\tilde{\theta}_{n,m}(\tau)$	Generalised and time warped, motion coordinate n
$\theta_{n,m}(k)$	Generalised, discretised, time warped motion coordinate n
Φ	The motion function vector
P	The function set coefficient parameters
O	The number of harmonic components
H	The model hyperparameters
N	The samples per second of discretised trajectories
S, S_0	Hyper-planes slopes and offset
ε_l	Constraint condition
$\triangle$	Sensed variation
δP	Motion correction vector

systems (no inputs) and producing in output the observed trajectory motion. Among these are:

- switching-linear-dynamic-systems (SLDS) [12] decompose complex human motions through a sequence of models, each one modelled with a linear time invariant discrete time (LTI-DT) system. The training algorithm of these systems is similar to the Hidden-Markov-Models's one, and relies only on motion observations. [13] adapted the algorithm to manage the dynamic nature of the processes. SLDSs decompose complex motions into more elementary chunks. However, this degree of autonomy has some drawbacks, such as the difficulty to check model properties in the presence of environmental perturbations that alter the model response;
- local models were introduced by Atkeson et al. [14]. Atkeson approximated the behaviour of a complex system with a set of simpler (local) models. Atkeson realised each model with a neural network and consequently trained them. Vijayakumar et al. [15] improved this approach and introduced a method known as Locally Weighted Projection Regression (LWPR). LWPR allows for on-line learning and higher dimension data handling.
- Gaussian Mixture Regressions (GMR) [11] identify the dynamic motion by using a set of probability distributions. GMRs use state space (SS) models. In such a representation, the motion dynamics is a multidimensional mathematical function. The learning algorithm extracts this function from the process observations. The algorithm uses example data to estimate the shape of the state descriptive function. The learning algorithm computes the derivatives of the trajectories to produce a scatter plot. Then the algorithm maximises the expectation of the computed plot. The algorithms use a mixture of multidimensional distributions and finds out the means and variances that maximise the data expectation. The use of derivatives in the GMR learning process introduces noise into the learning procedure. To improve the overall stability of the achieved model, Khansari-Zadeh and Billard [8] proposed a modified learning procedure that makes use of learning constraints to force the attractors of each linear subsystem to share the common final target point;
- Dynamic Motion Primitives (DMPs) [16] represent motion with a sequence of II-order attractor systems. An Expectation Maximisation (EM) tool [10] identifies the systems' parameters (centres, stiffness and viscosity coefficients). DMPs always generate stable systems. Further optimisation procedures can manipulate DMP models to match with given boundary conditions [15]. Recently, Reinforcement Learning [10] adapts DMP models for dynamic constraints [17]. This procedure operates after the model learning and introduces some distortions in the motion. DMP may create a stable guidance system from just one valid demonstration. The compensation for changes in the environment is provided through a computational algorithm and ignores any specific control strategy that is present in the training data.

Several other modelling procedures are presently employed in research, among these are: Gaussian-Process-Latent-Variable-Models (GP-LVM) [18], Gaussian-Process-Dynamical-Models (GPDM) [19], and Model-Predictive-Control Simulators (MPC-SIM) [20]. A broad review of them has been discussed in [21].

In this work, we present an analytical model that learns motions from a limited set of human demonstrations. The model makes use of parametric expansion-function series (ES) [22]. The ES structure only depends on the number of components the designer selects for learning according to clear rules. In contrast with similar approaches [23], the present work specifically focuses on the variability between tasks, as relevant information to be exploited during the learning. We will introduce two elements of novelty: the first relies in the ability of the learning algorithms to benefit from different, yet homogenous examples. The second is an adaptation tool that allows us to manage real-time changes on the programmed trajectory. The resulting mathematical models allow a fine degree of control on its boundary values so it can achieve long and complex robot motions.

As we will see, the algorithm relies on robust linear regression. This regression operates in the latent space defined by the ES coefficients. This learning is similar to Support-Vector-Machine and Kernel-Methods [24]. The major difference is the algorithm uses a predefined function set that eases the introduction of learning criteria (for example boundary conditions, energy of motions, etc.) and real-time processing.

2. Motivation

In literature, a wide set of motion models for human motion generation exist. Why should we start developing another model? We started working in a skill transfer context. This context requires us to capture and transfer the abilities between subjects [25,26]. We captured user motion from experts and created a dataset for training models. Multi-modal interfaces used such models to make novices practicing. We collected demonstrations from both experts and trainers of several fields, such as sport (rowing [27]), surgery (maxillo facial surgery [28]), industry (assembly and maintenance operation [29]), and entertainment (juggling [30]). We found that a common ground exists between all the above training fields [31]. Using interaction with experts and data analysis we decomposed complex motions into simpler motions [32]. The decomposition proceeds through semantic analyses and machine learning algorithms [33], such as Hidden Markov Models [34], and Probabilistic Neural Networks [35]. We clustered the simpler motions in classes considered to exhibit stationary properties (task/environment). These classes were chosen by taking into account the brain activities during the associated task control and in order to maximise the likelihood that the brain only focuses on a unique and stationary activity, such as limb coordination, point reaching or force control.

Given the above specification we looked for models that fit for the given tasks, and having the following features: