Proceedings of the 2011 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Beijing, China

Development of Learning Control in Robots

Hu Su, De Xu, and Yanlong Huang

State Key Laboratory of Intelligent Control and Management of Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China.

{hu.su, de.xu & yanlong.huang }@ia.ac.cn

***Abstract* - Learning control has been an active topic of research for several decades, and is of theoretical, as well as practical, significance. Current theories and developments in learning control are discussed. Following a brief introduction of the state as well as new progress on learning control, we give a detail review on the models and algorithms of the control policies developed recently which proved to be advantageous over previous approaches through experimental results. The related results and properties are presented. Then, several potentially developmental topics that are valuable to be further investigated are suggested. Finally, the conclusion remark is proposed.**

Index Terms - Learning control, dynamic motor primitive, learn by imitation, locally weighted regress, locally weighted projection regress.

I. INTRODUCTION

With the advent of anthropomorphic and humanoid robots, a large number of new challenges have been posed to the field of robotics and intelligent systems. Along with it, some intelligent control strategies have been developed recently which involve different application background. Neural network is provided with the learning capacity, that is, it is enabled to adjust the weight for every neuron. Such learning capacity is acquired by simulating functions of human brain. Moreover, fuzzy control, another intelligent control method, has achieved considerable development. This controlling method, built on the basis of fuzzy set, shows satisfying control effect by the means of simulating human beings' thinking. In addition, the fuzzy neural control based on the combination of neural network and fuzzy control has a wide application in industry production. But these control strategies are not adapted to the demands of the developments of robots, for failing to coordinate with frequent contacts of robots with an unknown environment. Besides, an important limitation to the application of robots to tasks in our daily lives is the combinatorial explosion of numbers of situations the robot may enter, for there are large numbers of sub-tasks, states, and other possible environments in which the robot must be enabled to operate, and possible exceptions can occur during execution.

Currently, still many robots try to follow precomputed trajectories. It is even impossible to control the robots efficiently. Thus, it is necessary to develop new kinds of controllers that can cope with changing environments and can be taught by unskilled human users.

In order to address the last issue, programming by demonstration (PbD) has emerged as a promising approach [1]. PbD covers methods by which a robot learns new skills through human guidance. Besides that, inspired by human behaviours and the structure of human brains, several approaches are established.

Firstly, motivated by the evidence that humans often rely on a set of motor primitives and use imitation as well as reinforcement learning when learning new skills, the idea has been developed which uses DMP as a general approach of representing control policies for basic movements.

Secondly, the fact can be concluded from actual experiences that human beings have the ability of reacting quickly to external stimulations which could be reinforced under repeated training. It is generally believed that people have this ability because a mapping relationship exists in human brains [36]. The algorithm, named as Locally Weighted Regression (LWR), is conceived to make the robot possess the self-learning ability and adapt to any complex environment so that the robot could react to external stimulations as quickly as human beings do.

Finally, with the advent of high dimensional input data, nonlinear function approximation remains a nontrivial problem, especially in incremental and real-time formulations. Statistical learning algorithms which fit nonlinear function globally are under development recently, such as Gaussian Process Regression (GPR) [1], Support Vector Machine Learning (SVML) [2], and Variational Bayesian for Mixture models (VBM) [3]. But these approaches are not the most suitable for on-line learning in high-dimensional spaces as they are mainly involved in batch data analysis and are not efficiently adjusted for incrementally arriving data in spite of the solid theoretical foundation that the approaches possess in terms of generalization and convergence. A method, named as Locally Weighted Projection Regression (LWPR), is presented which could cope with high dimensional inputs efficiently by using techniques of projection regression.

In this paper, current theories and recent development in learning control are discussed. Detailed explanation of models and algorithms of the control policies are also included. Section II introduces model and algorithm of dynamic motor primitive. In section III, Hidden Markov Model (HMM) and Gaussian Mixture Regression (GMR) are presented in detail. How to model the control policies using GMR and HMM is also discussed. Learning algorithms from only local information are provided in section IV. Finally, possible directions for future research of learning control are discussed.

II. MODEL AND ALGORITHM OF DYNAMIC MOTOR PRIMITIVE

It is reasonable make the assumption that movement generation is highly modular in terms of motor primitives (i.e. unit of movement). The motor primitives are modelled as solutions of respectively a dynamical system with a globally attractive fixed point and an oscillator. A formulation of the primitives with autonomous nonlinear differential equations is called DMP, whose time evolution creates smooth kinematics control policies.

The recent developed motor primitives based on dynamic systems has allowed both imitation and reinforcement learning to acquire new behaviours fast and reliable [2], [5], [6] ,[11]. Previous applications include a variety of different basic motor skills such as tennis swings [2], [5], baseball batting [7], [8], drumming [9], biped locomotion [10] and even in tasks with potential industrial application [12]. Moreover, a new method was presented to represent a demonstrated motion as an autonomous (time independent) non-linear first order ordinary differential equation (ODE) [24].

The basic idea in the original work is that motor primitives can be parted into two components, i.e., a canonical system and a transformed system for every degree of freedom *k* **[3]** with a one-way, parameterized connection such that one system drives the other (See Fig. 1).

Canonical system: $\tau\dot{z}$=h(z). The canonical system could be initialized by a first-order system as well as a second-order system. It acts as an adjustable clock or phase of the movement with state. The canonical system h drives the second component, the transformed systems. Transformation system: $\tau\dot{z}$ =g(x,z,w), for all considered degrees of freedom (DOFs) i, where z denotes the state of the canonical system

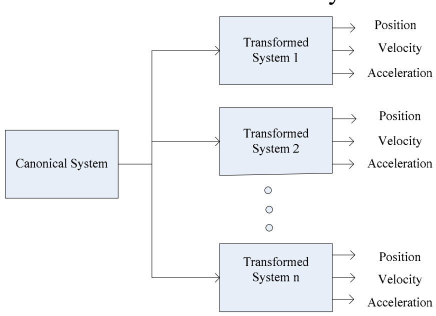

Fig. 1. Schematic illustration of an MP

and w the internal parameters for transforming the output of the canonical system [13]. Also, the variables of internal focus x is determined by the differential equations.

The transition from one movement segment to another could be state-triggered or time-indexed, in that way all trajectories (for each joint) could be generated through a unique set of differential equations. Therefore complex movements are generated through the superimposition and sequencing of simpler motor primitives generated by rhythmic and discrete unit generators.

A. Dynamical Systems for Trajectory Formation

Under the assumption of existence of two basic types of motor primitives, i.e., discrete [2-6] and rhythmic movements [3], [5], [11], we take the rhythmic movements as an example to explain the representation of DMP. In this section, it is supposed that the following rhythmic system has a stable limit cycle in terms of polar coordinates (Φ, r).

Transformation System:

$$\tau\dot{z} = \alpha_z[\beta_z(y_m - y) - z] \tag{1}$$

$$\tau\dot{y} = z + \mathrm{f} \tag{2}$$

Canonical System:

$$\tau\dot{\Phi} = 1 \tag{3}$$

$$\tau\dot{r} = -\mu(r - r_0) \tag{4}$$

The nonlinear function f is defined as follows:

$$\mathrm{f}(x, v, g) = \frac{\sum_{i=1}^{N}\psi_i\omega_i\tilde{\mathrm{v}}}{\sum_{i=1}^{N}\psi_i} \tag{5}$$

where $\Psi_i = \exp\{-h_i[\mathrm{mod}(\Phi, 2\pi) - c_i]^2\}, \tilde{\mathrm{v}} = [r\cos\Phi, r\sin\Phi]^T$, α_z, β_z are time constants, τ is a temporal scaling factor, a nd y, $\acute{y}$ correspond to the desired position and velocity generated by the policy as a movement plane. Additionally, the monotonic global convergence to g can be guaranteed with a proper choice of α_z and β_z and y_m is the anchor point for the oscillatory trajectory.

B. Learning From Demonstration

The mixture of primitives learns a desired trajectory from the demonstration by adjusting the set of weights, ω_i. Given a desired trajectory y_{demo} from the demonstration of a teacher, it is required that at the time when the final target is reached, the time dynamics (v, x) reaches (g, 0). Assuming the goal g is known, locally weighted regression is used to adjust the weights on line which minimizes the locally weighted error $J_i = \sum_t \psi_i^t (u_{des}^t - u_i^t)^2$, where $u_{des} = \dot{y}_{demo} - z$, and for each kernel function $\Psi_i^t, u_i^t = \omega_i v^t$ .A novel reinforcement learning algorithm, called policy learning by weighting exploration with the returns (PoWER), is presented [11] [13] ; experimental results has proved it to be advantageous over the algorithms in [14].

C. Modification

An improved modification of the original dynamic of the original DMP is presented in [1], which generalize movements to new targets without singularities and large accelerations. Then, the formalism is further extended to obstacle avoidance by adding an additional term to the differential equations which makes the robot steer around an obstacle.

D. DMP with Perceptual Coupling [13]

On the basis of the original DMP mentioned above, an external variable was taken into account which only affects the transformation system. This method allows continuous modification of the current state of the system by another variable. In this case, a modified dynamical system is defined as:

$$\tau\dot{\mathrm{z}} = \mathrm{h(z)} \tag{6}$$

$$\tau\dot{\mathrm{x}} = \hat{\mathrm{g}}(\mathrm{x}, \mathrm{y}, \overline{\mathrm{y}}, \mathrm{z}, \mathrm{v}) \tag{7}$$

$$\tau\dot{\bar{y}}=g(y,z,w) \tag{8}$$

where y denotes the state of the external variable, $\bar{y}$ the expected state of the external variable and $\dot{\bar{y}}$ its derivative. The concrete forms of functions h, $\bar{g}$ and $\hat{g}$ are presented in [13].

E. Properties

1) The trajectory converges to the goal point and automatically adapt to perturbations.

2) By setting τ, the duration of a movement could be modified without changing the movement trajectory.

3) The trajectories which have similar parameters ω_i are bracketed together, in that way all the trajectories could be classified.

III. Learn by Imitation

Most approaches to trajectory modelling estimate a time-dependent model, by either exploiting variants along with the concept of spline decomposition [16]-[18] or through statistical encoding of the time-space dependencies [19], [20]**.** Although such modelling methods are precise in the description of the actual trajectory, they have some problems in handling time variation which makes the methods sensitive to both temporal and spatial perturbations.

As an alternative, dynamical systems have been recently advocated as a powerful means of modelling robot motions [21]-[23], which considered modelling the intrinsic dynamics of motion.

A. Models

1) Hidden Markov Model (*HMM*)*:* HMM use a mixture of multivariate Gaussians to describe the distribution of the data. HMM encapsulate the transitions probabilities between the Gaussians [25] [26].

Let {Π, A, B} be, respectively, the initial state distribution, the transition probabilities between the states, and the multivariate output data distributions [27] [28]. The parameters {Π, A} are learned using Baum-Welch algorithm **[27]**, which is a variant of EM algorithm [29] and set $B=\{\mu_k, \Sigma_k\}_{k=1,\cdots,K}$, where $\{\mu_k, \Sigma_k\}_{k=1,\cdots,K}$ are the state distributions.

Once trained, the HMMs can be used to recognize gestures. In the experiments, this is used to decide whether a new demonstration belongs to the same task. The forward – algorithm was used to measure the similarity between a new gesture and the ones encoded in the model.

2)Gaussian Mixture Regression (*GMR*)*:* GMR does not model the regression function directly, but models a joint probability density function of the data, then derives the regression function from the density model.

A data set of *N* data points of *D* dimensions is encoded in a Gaussian Mixture Model (GMM) [25], [26], input vectors and output vectors are denoted as $\{\xi_t^I\}_{t=1}^N$, $\{\xi_t^O\}_{t=1}^N$ and $\xi=[\xi^I, \xi^O]$. The probability ξ that a data point belongs to GMM is defined by

$$p(\xi)=\sum_{k=1}^{K}\pi_k N(\xi;\mu_k,\Sigma_k) \tag{9}$$

where π_k are prior probabilities and $N(\mu_k, \Sigma_k)$ are Gaussian distributions defined by centers μ_k and covariance matrices Σ_k.

$$\mu_k=\begin{bmatrix}\mu_k^I\\ \mu_k^O\end{bmatrix}, \Sigma_k=\begin{bmatrix}\Sigma_k^I & \Sigma_k^{IO}\\ \Sigma_k^{OI} & \Sigma_k^O\end{bmatrix}$$

For a given input variable ξ^I and a given Gaussian distribution *k*, the expected distribution of ξ^O is defined by

$$p(\xi^O\left|\xi^I,k\right.)\sim N(\hat{\xi}_k,\hat{\Sigma}_k) \tag{10}$$

$$\hat{\xi}_k=\mu_k^O+\Sigma_k^{OI}(\Sigma_k^I)^{-1}(\xi^I-\mu_k^I),\ \hat{\Sigma}_k=\Sigma_k^O-\Sigma_k^{OI}(\Sigma_k^I)^{-1}\Sigma_k^{IO}$$

The expected distribution of ξ^O, when ξ^I is known, can be estimated as:

$$p(\xi^O\left|\xi^I\right.)\sim\sum_{k=1}^{K}h_k N(\hat{\xi}_k,\hat{\Sigma}_k)=N(\hat{\xi},\hat{\Sigma}) \tag{11}$$

More details please refer to [30].

B. Control Strategy Design

Fig. 2 gives an overview of the input-output flow through the complete model, three processes are included in the model: probabilistic data encoding, determining the task constraints, optimal trajectory generation.

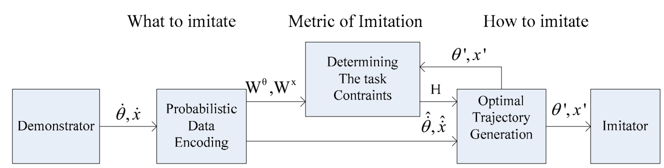

Fig. 2. Information flow across the complete system

1) Trajectory-based Control [25], [26], [30], [31]*:* Let $\hat{\dot{\theta}}$ and $\hat{\dot{x}}$ be respectively the desired joint angle velocities and the generalized end-effector velocities in Cartesian space, and let $\dot{\theta}$ and $\dot{x}$ be the candidate velocities for reproducing the motion. The cost function *H* is defined as follows which measures the variation of constraints and of dependencies across the variables over time. An optimal controller can be obtained by solving the constrained optimization problem

$$\min\ H=(\hat{\dot{\theta}}-\dot{\theta})^T W^\theta(\hat{\dot{\theta}}-\dot{\theta})+(\hat{\dot{x}}-\dot{x})^T W^x(\hat{\dot{x}}-\dot{x}), s.t.\ \dot{x}=J\dot{\theta}$$

where *J* is the Jacobian matrix at posture θ_t, $W^\theta \in R^{nxn}$ *and* $W^x \in R^{mxm}$ are semi-definite positive diagonal matrices serving as coefficient indicating the respective influence.

Using Lagrange multipliers, a solution can be obtained:

$$\dot{\theta}=(W^\theta+J^+W^xJ)(J^+W^x\hat{\dot{x}}+W^\theta\hat{\dot{\theta}}) \tag{12}$$

where J^+ denotes the Moore-Penrose pseudo-inverse of the Jacobian matrix. An alternative representation of the above equation was presented in [32] that have been proved to be advantageous from an implementation perspective. The joint angle trajectories are finally found by using the relation $\theta(t) = \theta(t-1) + \dot{\theta}$.

More details for efficient implementation, such as the reduction of dimensionality, the selection of the optimal number of GMM components K can be obtained in [26].

2) Tracking-based Control [33]-[35]: A desired velocity is estimated through GMR. Given the current position, a velocity command is estimated iteratively to control the system. That is, the current position x is the input vector, the same parameter as ξ^I referred in GMR and, the desired velocity $\hat{x}$ is the output vector, the same parameter as ξ^O referred in GMR.

$$\hat{x} = \sum_{i=1}^{K} h_i(\xi^I)[\mu_i^O + \Sigma_i^{OI}(\Sigma_i^I)^{-1}(\xi^I - \mu_i^I)] \tag{13}$$

where, $h_i(\xi^I)$ represents the HMM forward variable, $h_i(\xi^I)$ is initialized by $h_{i,t}(\xi^I) = \pi_i N(\xi_1^O; \mu_i^O, \Sigma_i^O) / \sum_{i=1}^{K}[\pi_i N(\xi_1^O; \mu_i^O, \Sigma_i^O)]$ [20], [21] and corresponds to the probability of observing the partial sequence $\{\xi_1^O, \xi_2^O, \cdots, \xi_t^O\}$ and of being in state i at time t.

Similarly, a target position is retrieved from the estimate of dynamics of motion. Given the current velocity, a position command is estimated iteratively to control the system. The current velocity $\dot{x}$ is the input vector while the desired velocity $\hat{x}$ is the output vector.

$$\hat{x} = \sum_{i=1}^{K} h_i(\xi^I)[\mu_i^I + \Sigma_i^{IO}(\Sigma_i^O)^{-1}(\xi^I - \mu_i^O)] \tag{14}$$

The acceleration command is determined by:

$$\ddot{x} = (\hat{\dot{x}} - \dot{x})\kappa^v + (\hat{x} - x)\kappa^p \tag{15}$$

where κ^v and κ^p are gain parameters similar to damping and stiffness factors. Particular forms of the velocity gain κ^v and the position gain κ^p are defined in [33], [34].

C. Properties

1) Trajectory-based Control: The approach presents an architecture of solving the problem for extracting the constraints of given tasks in a programming by demonstration framework and the problem of generalizing the acquired knowledge to various contexts. Under the constraints which are essential for reproducing the demonstration, the trajectory is obtained which optimizes the cost function composed of imitation metric and the constraints. The approach can be extended to different robot architecture which is advantageous over the previous approaches.

2). Tracking-based Control: The techniques that have been proposed over the years suffer from explicit time dependency which makes them sensitive to both temporal and spatial perturbations. Using HMM allowed us to get rid of the explicit time dependency, by encapsulating precedence information within the statistical representation.

IV. Learn from only Local Information

A. Locally Weighted Regression

LWR was first proposed as an approach combining the simplicity of linear least squares regression with the flexibility of nonlinear regression [42], which was then applied to robot control [43], [44].

LWR can be applied in a much broader context. Global learning methods can often be improved by localizing them using locally weighted training criteria [45], [46]. Up to now, the application of LWR includes learning the forward model which is necessary in studying the task-space control [40], table tennis robot controlling [41], tracking of blood pressures from childhood to adulthood [47] and so on.

1) Memory-based Learning: Actually, locally linear model have been brought into wide use, as they accomplish a favourable compromise between computational complexity and quality of result.

$$\hat{y}_k = x^T b_k + b_{0,k} = \tilde{x}^T \beta_k,\ \tilde{x} = (x^T, 1)^T \tag{16}$$

where β_k denotes the parameters of the locally linear model with k training points and $\tilde{x}$ a compact form of the center-subtracted, augmented input vector to simplify the notation. Algorithm can be implemented as follows [37],[38]：

Step 1--Initially, experiences are simply stored in the memory.
Step 2--To answer any particular query, a weighted linear regression is performed. Some parameters which are referred as "fit parameters" in the following are identified to calculate a distance metric and the weighting function, and stabilize the solution.
Step 3--Using cross validation, the fit parameters could be updated.

2) Incremental Learning: The receptive field-based learning system (RFWR) generates locally linear models in each receptive field and blends them for prediction. Given a query, the average $\hat{y}$ of outputs of all activating models at the query is used as the prediction of the query. Without loss of generality, the k-th receptive field which is activated under the presented query is considered in this section.

$$\hat{y} = (\sum_{k=1}^{K} w_k \hat{y}_k) / (\sum_{k=1}^{K} w_k) \tag{17}$$

$$\omega_k = \exp[-\frac{1}{2}(x - c_k)^T D_k (x - c_k)], D_k = M_k^T M_k$$

The positive definite distance metric D_k determines the size and shape of the receptive field. M_k is an upper triangular matrix which ensures that D_k is a positive definite. In (17), $\hat{y}_k$ is the prediction using the linear model in k-th receptive field, $c_k \in R^v$ is the center of k-th receptive field.

The learning algorithm of RFWR determines the parameters c_k, M_k and β_k for each receptive field independently, i.e., without any information about the other receptive fields. RFWR adds and prunes receptive fields as needed, such that the number of receptive fields, K, will automatically adjust to the learning problem [4].

Given a training point $(\tilde{x}, y)$, the learning process is defined as follows:

Step 1--Incremental updating of β**:**

$$\beta_k^{n+1} = \beta_k^{\ n} + \omega_x p^{n+1} \tilde{x} e_{cv}^T \tag{18}$$

where $p^{n+1} = \frac{1}{\lambda}(p^n - \frac{p^n \tilde{x}\tilde{x}^T p^n}{\lambda / \omega_x + \tilde{x}^T p^n \tilde{x}})$, $e_{cv} = (y - \beta_k^{\ n^T} \tilde{x})$

The equation includes a forgetting factor λ in order to gradually cancel the contributions from previous data points.

Step 2--Incremental updating of M**:** Define locally weighted leave-one-out cross validation error as follows:

$$J = \frac{1}{W}\sum_{i=1}^{p}\frac{\omega_i \left\| y_i - \hat{y}_i \right\|^2}{(1-\omega_i \tilde{x}_i^T p \tilde{x}_i)^2} + \gamma \sum_{i,j=1}^{n} D_{ij}, W = \sum_{i=1}^{p} \omega_i \tag{19}$$

The notation $\hat{y}_i$ denotes the prediction of i-th data point which is calculated from the learning system. To minimize the cross validation by adjusting M by gradient descent with learning rate α:

$$M^{n+1} = M^n - \alpha \cdot (\partial J / \partial M) \tag{20}$$

Step 3-- Adding and Pruning Receptive Fields: A new receptive field is created if a training sample $(\tilde{x}, y)$ does not activate any of the existing receptive field by more than a threshold w_{gen}. The parameters of the new receptive field are set to a manually chosen default value.

A receptive field is pruned if it overlaps too much with another receptive field. This effect is detected by a training sample activating two receptive fields simultaneously more than w_{prune}. The receptive field with the larger determinant of the distance metric D is pruned.

B. *Locally Weighted Projection Regression* (*LWPR*)

Under the assumption that data is characterized by locally low dimensional distributions [51], LWPR copes with high dimensional inputs efficiently. Currently, more and more researchers focus on solving various practical problems encountered in the physical world by using LWPR. LWPR has been applied in visual servoing which is used in learning the control of the low-cost robotic arm [62], [63]. And the suitability of LWPR to solve two problems: the prediction of littoral drift and that of scour downstream of a flip bucket spillway is assessed [59].

Locally weighted projection regression is a new algorithm for incremental nonlinear function approximation in high dimensional spaces with redundant and irrelevant input dimensions. LWPR copes with high dimensional inputs by using techniques of projection regression (*PR*) [52-56]. A variety of linear dimensionality reduction techniques are presented in related literature for the purpose of nonlinear function approximation, such as locally weighted factor analysis [52], principal component regression [53], partial least squares[54], [55] and so on. And sigmoidal neural networks can also be conceived as a method of projection regression [56]. In addition, (Vijayakumar and D'Souza) presented a novel projection regression technique, named covariant projection regression (LWCPR) [51] which has been proved that it accomplished excellent regression results with relatively few projections.

1) Models: With the assumption that the data generating model for the regression problem has the standard form $y = f(x) + \varepsilon$, (where $x \in R^n$ is a n-dimensional input vector, the noise term ε has mean zero, $E\{\varepsilon\}=0$ and the output is one-dimensional).

The prediction $\hat{y}$ for a query point x is built from the normalized weighted sum of the individual prediction $\hat{y}_k$ of all receptive fields, i.e., as shown in Eq. (17).

All input vectors are summarized in the rows of the matrix $X=[x_1, x_2, \cdots, x_P]^T$, the corresponding outputs are the elements of the vector y. P is the number of training data and N is the dimensionality of the input data. All the PR techniques considered here project the input data X onto k orthogonal directions [51], [57], along which they carry out univariate linear regressions.

2) The Complete LWPR Algorithm [51], [59], [60]: The approach of gradient descent updating of D is the same as step 2 and adding and pruning receptive fields is the same as step 3 of incremental learning. The initial number of projections is set to $R=2$.The algorithm determines whether R should be increased by recursively keeping track of the mean-squared error as a function of the number of projections included in a local model. If the MSE at the next projection does not decrease more than a certain percentage of the previous, i.e., $MSE_{i+1}/MSE_i > \Phi$, $\Phi \in [0, 1]$ the algorithm will stop adding new projections locally. More details for the algorithm please refer to [58], [59], [60].

C. *Properties*

LWPR can operate in very high dimensional space successfully and efficiently. If the input data is locally statistically independent and is approximately locally linear, LWPLS will find an optimal linear approximation for the data with a single projection. The major drawback of LWPR in its current form is the need for gradient descent to optimize the local distance metrics in each local model, and the manual tuning of a forgetting factor as required in the learning algorithms

LWR avoids the difficult problem of finding an appropriate structure for a global model. This approach is suitable for real time on line robot learning because of the fast incremental learning and the avoidance of negative interference between old and new training data. However, practical implementations require dealing with various difficult problems, such as inadequate amounts of training data, filtering of noise, and so on [38]. As a disadvantage to the incremental learning algorithm described above, an ever increasing number of receptive fields will be required to represent the approximated function.

V. DISCUSSIONS

A. Analysis

HMM approach shares many characteristics with the DMP approach. However, compared with DMP, HMM allows to be provided with partial demonstrations, which is a very important characteristics for the teaching interaction. DMP and HMM are both of time-independency approaches; DMP gets rid of the explicit time dependency by modelling the intrinsic dynamic of motion using two sets of differential equations, while HMM encapsulates the time variable within the statistical representation. However, the HMM method has the disadvantage that its stability relies on the proper choice of the gains in (15). In GMR, time is considered as an explicit variable to encode the motion. And it is possible to extract the constraints of given tasks automatically using GMR. However, although GMR can describe the trajectory precisely, it is sensitive to various external perturbations. In addition, LWR approximates a function from samples of the function's inputs and outputs avoiding the difficult problem of finding an appropriate structure for a global model, but many experiences are needed and its performance is not guaranteed.

B. Development Tendency

In recent years, learning control has made a great deal of progress in both research and application fields. But a complete theory system is not estimated that can guide design practice of controller. The learning control polices mentioned in this paper have diverse backgrounds and applications, thus can be developed in different directions.

Firstly, DMP could be applied to a full-body humanoid robot in the future, so that different robots could be developed to complete various functions needed in human life.

Secondly, as to imitation learning using HMM and GMR, future works could focus on extending the proposed approach to motion with more complex dynamics and to controllers that can adapt robustly to various external perturbations. In addition, such approaches should be developed to allow users to provide partial demonstrations to the robot. And the control of the legs of robots also calls for more attention which is very important for keeping balance and accomplishing particular tasks for the whole robots.

Finally, although LWR & LWPR have been widely applied to the robot controlling, it seems necessary to develop new data management algorithms, including principled ways to forget or coalesce old data, and compactly represent high dimensional data clouds and so on.

But also, some methods could be combined to get better performance in the future, such as DMP or HMM could be combined with LWPR which is useful to deal with high-dimensional inputs and HMM can also be used to tune the parameters of LWPR automatically which are manually tuned in previous methods.

VI. Conclusions

We survey some ongoing and past activities in robot learning to assess where the field stands and where it is going. Three types of control policies developed recently are discussed which have been proved to be effective in experiments as well as the modification for some specific tasks. What's more, detailed models and learning algorithms are also presented. Finally, properties of the control strategies mentioned in the paper and possible future research directions for learning control are given.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61075035 and National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2011ZX 04013 - 011.

References

[1] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. on Robot. and Automation*, Japan, 2009, pp.2587-2592.

[2] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots," in *Proc. IEEE Int. Conf. on Robot. , Autom.,* Washington, DC, 2002, pp.1398-1403.

[3] S.Schaal, J.Peters, J.Nakaishi, and A.Ijspeert, "Learning Movement Primitives," *Robot. Research*, STAR 15, pp.561-572, 2005.

[4] A.J.Ijspeert, J.Nakanishi, and S.Schaal, "Trajectory Formation for Imitation with Nonlinear Dynamical Systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Syst.* , Hawaii, 2001, pp.752-757.

[5] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances Neural Information Processing Systems*, Vol. 15, Cambridge, MA: MIT Press, 2003, pp.1547-1554.

[6] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control – a unifying view," *Progress in Brain Research*, vol. 165, no.1, pp.425-445, 2007.

[7] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Pro. IEEE Int. Conf. on Intelligent Robots and Syst.*, Beijing, China, 2006, pp.2219-2225.

[8] J. Peters and S. Schaal, "Reinfocement learning of motor skills with policy gradients," *Neural Networks* ,vol. 21, iss . 4, pp.282-297, 2008.

[9] D. Pongas, D, A. Billard, and S. Schaal, "Rapid synchronization and accurate phase-locking of rhythmic motor primitives," in *Pro. IEEE Int. Conf. on Intelligence Robot. and Syst.*(*IROS*) , 2005, pp.2911-2916.

[10] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schall, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robot. and Autonomous Syst.*, vol. 47, no, 2-3, pp.79-91, 2004.

[11] J. Kober and J. Peters, "Imitation and Reinforcement Learning," *IEEE Robot. Autom. Mag*., vol. 17, iss. 2, pp.55-62, 2010.

[12] H. Urbanek, A. Albu-Schaffer, and P. van der Smage, "Learning from demonstration repetitive movements for autonomous service robotics," in *Proc. IEEE Int. Conf. on Robots and Syst.* (*IROS*), 2004, pp.3495-3500.

[13] J. Kober, B. Mohler, and J. Peters, "Learning perceptual Coupling for Motor Primitives," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS),* Nice, France, 2008, pp.834-839.

[14] J. Kober and J. Perters, "Policy search for motor primitives in robotics," *Mach. Learn.* , 2009.

[15] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robot.,* B. Siciliano and O. Khatib, Eds. New York: Springer-Verlag, 2008, ch.59.

[16] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths", *Robot. and Autonomous Syst*., vo. 11, no. 2, pp. 113-127, 1993.

[17] H. Friedrich, S. Muench, R. Dillmann, S. Bocionek, and M. Sassin, "Robot programming by demonstration: Supporting the induction by human interaction," *Machine Learning*, vol. 23, no.2, pp.163-189, 1996.

[18] J. Aleotti and A. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," *Robot. and Autonomous Syst.* , vol. 54, no. 5, pp.409-413, 2006.

[19] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robot.*, vol. 23, no. 15, pp.2059-2076, 2009.

[20] M. Muehlig, M. Gienger, S. Hellbach, J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *Pro. IEEE Conf. on Robot. and Autom.*, Japan, 2009, pp.1177-1184.

[21] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Phil. Trans. R. Soc. Biological Sciences* (*The Royal Society*), vol. 358, no.1431, pp.537-547, 2003.

[22] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Pro. Int. Conf. on Robot. and Autom.* (*ICRA*), 2002.

[23] D. Grimes, D. Rashid, and R. Rao, "Learning nonparametric models for probabilistic imitation," in *Advances in Neural Information Proc. Syst.* , 2006.

[24] M. S. Khansari-Zadeh and A. Billard, "BM: An Iterative Algorithm to Learn Stable Non-Linear Dynamical Systems with Gaussian Mixture Models," in *Proc. IEEE Conf. on Robot. and Autom.* (*ICRA*), Alaska, USA, 2010, pp.2381-2388.

[25] S. Calinon, F. Guenter, and A. Billard, "On Learning the Statistical Representation of a Task and Generalizing it to Various Contexs," in *Proc. IEEE Conf. on Robot. and Autom.* , Florida, 2006, pp.2978-2983.

[26] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing, and Generalizing a Task in a Humanoid Robot," *IEEE Trans. on Syst., Man, and Cybernetics*, vol. 37, no. 2, pp.286-298, 2007.

[27] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in *Proc. of the IEEE*, vol. 77, no. 2, pp.257-286, 1989.

[28] D. Kulic, W. Takano, and Y. Nakamura, "Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains," *The Int. Journal of Robot. Research,* vol. 27, no. 7, pp.761-784, 2008.

[29] Z. Ghahramani and M. J. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Proc. Systems 6*, Morgan Kaufmann Publisher, San Francisco, CA, 1994.

[30] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space," *Advanced Robotics*, vol. 23, no. 15 , pp.2059-2076, 2009.

[31] Hersch, M., Guenter, F., Calinon, S., Billard, A., "Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations", *IEEE Trans. on Robotics*, vol. 24, no. 6, pp.1463-1467, 2008.

[32] M. Hersh, and A. Billard, "Reaching with multi-referential dynamical systems," *Autonomous robots*, vol. 25, no. 1-2, pp.71-83, 2008.

[33] S. Calinon, F. D'halluin, D. Caldwell, and A. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Proc. 9th IEEE-RAS Int. Conf. on Humanoid Robots*, Paris, France, pp.582-588, 2009.

[34] S. Calinon, E. Sauser, A. Billard, and D. Caldwell, "Evaluation of a probabilistic approach to learn and reproduce gestures by imitation," in *Proc. IEEE Int. Conf. on Robotics and Autom.* (*ICRA*), Alaska, USA, 2010, pp.2671-2675.

[35] S. Calinon, F. D'halluin, E. L. Sauser, D. Caldwell, and A. Billard, "Learning and Reproduction of Gestures by Imitation," *IEEE Roboti. & Autom. Mag.*, vol. 17, iss. 2, pp.44-54, 2010.

[36] Q. Rui, Z and W. Hu "Simulation of Ping-pang trajectory prediction using LWR," *Robot*, Vol. 20, No. 5, pp.373-377, 1998.

[37] C. G. Atkeson, "Using Locally Weighted Regression for Robot Learning," in *Proc. IEEE Int. Conf. on Robot. and Autom*., Califomia, 1991, pp.958-963.

[38] C. G. Atkeson, A.W.Moore, and S.Schaal, "Locally weighted learning," *Artificial Intelligence Review,* vol. 11, no. 1-5, pp.11-73, 1997.

[39] S.Schaal and C.G..Atkeson, "Constructive Incremental Learning From Only Locally Information," *Neural Computation*, vol. 10, no. 8, pp. 2047-2084, 1998.

[40] A. C. Lammert, L. Goldstein, and K. Iskarous, "Locally-Weighted Regression for Estimation the Forward Kinematics of a Geometric Vocal Tract Model," in *Proc.11th Annual Conf. Int. Speech Communication Association*, Japan, 2010, pp.1604-1607

[41] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, "A Learning Approach to Robotic Table Tennis," *IEEE Trans. on Robot.*, vol. 21, no. 4, pp.767-771, 2005.

[42] W. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *American Statistical Association*, vol. 74, no. 368, pp.829-836, 1979.

[43] C. G. Atkeson, "Using local models to control movement," in *Advances in Neural Information Proc. Syst. 2*, pp.316-323, 1990.

[44] A. Moore, "Fast, robust adaptive control by learning only forward models," in *Advances in Neural Information P*roc. Syst. 4, San Francisco, CA, USA: Morgan Kaufmann, 1992, pp.571-579.

[45] V. Vapnik, "Principles of risk minimization for learning theory," in *Advances In Neural Information Proc. Systems 4*, Morgan Kaufman, San Mateo, CA, 1992, pp.831-838.

[46] V. Vapnik and L. Bottou, "Local algorithms for pattern recognition and dependencies estimation," *Neural computation* vol. 5, no. 6, pp.893-909, 1993.

[47] X. Chen and Y. Wang, "Tracking of Blood From childhood to Adulthood," *American Heart Association*, pp.3171-3180, 2008.

[48] C. K. I. C.Williams and C. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Proc. Systems 8*. MIT Press, 1996, pp.514-520.

[49] A. J. Smola and B. Scholkopf, B. "A tutorial on support vector regression," *Statistics and Computing*, *14*, pp.199-222, 2004.

[50] Z. Ghahramani and M. Beal, "Variational inference for Bayesian mixtures of factor analysers," in *Advance in Neural Information Proc. Systems 12*, MIT Press, 2000, pp.449-455.

[51] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental Online Learning in High Dimensions," *Neural Computation*, vol.17, no.12, pp.2602-2634, 2005.

[52] B. S. Everitt, "An introduction to latent variable models," *Chapman and Hall*, London, 1984.

[53] S. Vijayakumar and S. Schaal, "Local Adaptive Subspace Regression," in *Neural Proc. Letters 7*, 1998, pp.139-149.

[54] I. E. Frank and J. H. Fridman "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, vol.35, no.2, pp.109-135, 1993.

[55] H. Wold, "Perspectives in probability and statistics," 1975, Chapman Hall.

[56] S. E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture," in *Advances in Neural Information Proc. System* 2, 1990.

[57] S. Vijayakumar and S. Schaal, "Locally Weighted Projection Regression: An o (n) Algorithm for Incremental Real Time Learning in High Dimensional Space," in *Pro. Int. Con. on Mach. Learn.*, 2000, pp. 1079-1086.

[58] S. Klanke, S. Vijayakumar, and S. Schaal, "A Library for Locally Weighted Projection Regression," *Journal of Mach. Learn. Research 9*, pp.623-626, 2008.

[59] M. Agarwal, M. Goyal, and M. O. Deo "Locally weighted projection regression for predictiong hydraulic parameters," *Civil Engineering and Environmental Systems*, vol.27, no.1, pp.71-80, 2010.

[60] S. Vijayakumar, S. Schaal, "Fast and Efficient Incremental Learning for High-dimensional Movement Systems," in *Proc. of 2000 IEEE Int. Conf. on Robot. & Autom.* (*ICRA*), San Francisco, 2000, pp.1894-1899.

[61] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Real-Time Robot Learning With Locally Weighted Statistical Learning," in *Proc. of 2000 IEEE Int. Conf. on Robot. & Autom.* (*ICRA*), San Francisco, 2000, pp.288-293.

[62] F. Larsson, E. Jonsson, and M. Felsberg, "Simultaneously learning to recognize and control a low-cost robotic arm," *Image and Vision Computing*, vol. 27, no. 11, pp.1729-1739, 2009.

[63] F. Larsson, E. Jonsson, M. Felsberg, "Visual servoing for floppy robots using LWPR," in *Proc. Int. Workshop on Robot., Mathematics*, 2007.