The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

A machine learning approach to 9-DOF robotic arm control

†S. Nishioka, †S. Maeda, ‡Y. Nakamura, †T. Ueno, ‡H. Ishiguro and †S. Ishii

†Graduate School of Informatics Kyoto University
†(Tel : +81-774-38-3938; Fax : +81-774-38-3941)
†Email :{nishioka-s, ichi, tsuyos-u, ishii}@sys.i.kyoto-u.ac.jp
‡Graduate School of Engineering Science, Osaka University
‡(Tel : +81-6-6879-7970; Fax : +81-6-6850-6360)
‡Email :{nakamura@is, ishiguro@}.sys.es.osaka-u.ac.jp

Abstract: In this study, we propose a new control strategy for a arm robot system which has 9-DOF actuated by McKibben artificial muscles. Since the dynamics of the McKibben actuators are complex due to the dependence on the internal pneumatic pressure, it is difficult to control the robot by solving the inverse kinematics according to the motion equations in real-time computing. Also, the state of each link of the arm cannot be determined uniquely from the position of the hand due to the ill-posedness in the arm posture. To resolve these difficulties, we introduce a notion of imitation, that is, 1) a human tutor taught the robot several patterns of exemplary trajectories by moving the robotic arm directly, so that the robot's forward model can learn how the robot dynamics follow those exemplary trajectories. Then, 2) the exemplary trajectories are pursued by a controller that has learned the relationship between the control signal and the consecutive arm states before and after the control signal is applied during the free control phase prior to the real-time control.
Keywords: Motion Planning, Inverse Kinematics, Robotics, Artificial Muscle

I. Introduction

In recent years, studies of robots using artificial muscles have been actively carried out, because of naturalness of produced motions. McKibben artificial muscle is one of such artificial muscles, which is actuated by pneumatic power. The McKibben actuator is drawing an interest because it realizes flexible motions like human muscles and of light weight and reasonable cost [1].

The improvement in the servo system of the McKibben actuator, which has been brought by the recent large computation power, contributes to extending the application field of the McKibben actuator [1]. However, several inherent difficulties in the McKibben actuator hinder further practical use. One difficulty comes from the variation in the control valve; it causes large individual deviations in the degree of expansion and shrinkage of the actuator. Second, it does not necessarily emit the same control torque for a specific control input because its tension is dependent on the internal air pressure of the actuator, that is, there is an internal state. Moreover, when there are multiple actuators, they could physically interfere with each other. Therefore, the development of the control method that resolves these difficulties has been desired in order to allow the McKibben actuators to be used more in practical situations.

In this study, we propose a machine-learning approach to controlling a 9-DOF human-like robotic arm with 26 McKibben actuators, so that it follows the exemplary trajectories taught by a human tutor in advance. In particular, we aim to control the robot to reproduce handshake gesture. To realize natural movements, we do not set a rigid target trajectory, but let the controller perform a real-time motion planning in a flexible manner to the robot state. The motion planning is done by a forward model implemented as a Gaussian Process (GP) regressor [2], so that the generated trajectory resembles one of the exemplary trajectories. The generation of control signals to pursue the target trajectory is realized by an inverse model implemented as a Normalized Gaussian network (NGnet) [3], which learns the relationship between the control signal and the consecutive arm states before and after the control signal is applied during the free control phase. A statistical combination of these two models is expected to allow us to avoid the difficulty in solving the complex motion equations of the 9-DOF robot arm system actuated by McKibben actuators and to realize human-like natural movements.

II. Systems

1. Robotic Arm

Figure 1 shows the appearance of the robot arm system we used in this study, which imitates the human right arm. It has five joints and six links which are connected to and controlled by 26 McKibben actuators. A McKibben actuator is able to generate a contractile force axially when compressed air is put it in.

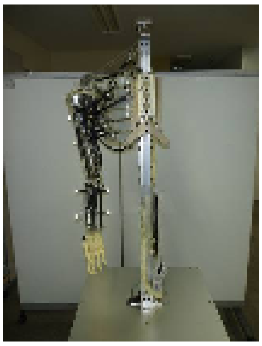

Fig.1. Appearance of the robotic arm

2. Observations and Control

The 3-D arm positions are monitored by a motion capture device (MAC3D system, Motion Analysis Corporation) with the sampling frequency of 200 Hz. Ten markers are attached so that the motion capture device can monitor all the positions of the 9 links. Since each marker position is represented as a three-dimensional vector, the state vector s_t that indicates all the positions of the ten markers becomes a 30-dimensional vector at each time t. During the imitation phase, in which a human tutor directly manipulates the robot arm like a handshake, exemplary trajectories of the manipulated robot arm, D, are also monitored by the motion capture device.

In addition to the link positions, the internal air pressure of each McKibben actuator is monitored. Since there are 26 McKibben actuators, the internal air pressure of all the actuators at time t are represented by a 26-dimensional vector P_t.

The robotic arm is controlled by a servo system; for each McKibben actuator, the target air pressure P_{gt} is provided as the control signal and the servo system works to realize the target pressure instantaneously.

III. Controlling method

After the robot controller has been trained based on the set of exemplary trajectories, D, the robot arm is controlled in real time by sequentially estimating the target air pressure P_{gt} at time *t*, given the current state s_t, the previous state s_{t-1}, and the current internal state P_t. This estimation is done by

$$\begin{aligned} & p(P_{gt} \mid P_t, s_t, s_{t-1}, D) \\ &= \int p(P_{gt} \mid P_t, s_t, s_{t+1}) p(s_{t+1} \mid s_t, s_{t-1}, D) ds_{t+1} \\ &\approx p(P_{gt} \mid P_t, s_t, \hat{s}_{t+1}) \end{aligned} \tag{1}$$

where $\hat{s}_{t+1} = \arg\max_{s_{t+1}} p(s_{t+1} \mid s_t, s_{t-1}, D)$. To solve the integral in the second line of eq. (1), the 'forward' model $p(s_{t+1} \mid s_t, s_{t-1}, D)$ is approximated by a Dirac's delta function $\delta(s_{t+1} - \hat{s}_{t+1})$ whose center is the MAP estimate of the forward model. Although the probabilistic distribution can provide rich information to the controller, it is assumed in this study that the controller emits the simplest mode signal, $\hat{P}_{gt} = \arg\max_{P_{gt}} p(P_{gt} \mid P_t, s_t, \hat{s}_{t-1}, D)$. Then, our task is to identify the two probability distributions, the 'inverse' model $p(P_{gt} \mid P_t, s_t, s_{t+1})$, and the forward model $p(s_{t+1} \mid s_t, s_{t-1}, D)$, through the training.

1. Gaussian Process (GP)

In the imitation phase, the forward model $p(s_{t+1} \mid s_t, s_{t-1}, D)$ is identified based on the exemplary trajectories D. We assume that in the forward model, each dimensionality (each marker position) is independent, i.e., $p(s_{t+1} \mid s_t, s_{t-1}, D)$ is factorized as $p(s_{t+1} \mid s_t, s_{t-1}, D) = \prod_i p(s_{t+1,i} \mid s_t, s_{t-1}, D)$ where $s_{t+1,i}$ is the *i*th element of s_{t+1}. Each constituent model $p(s_{t+1,i} \mid s_t, s_{t-1}, D)$ is represented by a non-parametric probabilistic model, Gaussian Process (GP) regressor [2].

Let $\mathbf{x}$ and y denote an input vector and an output from a GP regressor, respectively; $\mathbf{x}$ is a concatenation of the state vectors s_t and s_{t-1}, and y is a certain element of s_{t+1}. Using these notations, GP estimates $p(y \mid \mathbf{x}, D)$ as $p(y \mid \mathbf{x}, D) = N(y \mid m(\mathbf{x}; D), \sigma^2(\mathbf{x}; D))$, where $N(\cdot \mid m, \sigma^2)$ denotes a Gaussian distribution with mean m and variance σ^2.

When there are T series of input and output, $\mathbf{u} = \{\mathbf{u}_1, \cdots, \mathbf{u}_T\}$ and $\mathbf{v} = \{v_1, \cdots, v_T\}$, respectively, where $\mathbf{u}_i = [s_{t'}, s_{t'-1}]$ and $\mathbf{v}_i = s_{t'+1}$ in the set of exemplary trajectories D, the mean and the variance are given by $m(\mathbf{x}; D) = \mathbf{k}^T C^{-1} \mathbf{v}$ and $\sigma^2(\mathbf{x}; D) = c - \mathbf{k}^T C^{-1} \mathbf{k}$, respectively. Here, $\mathbf{k}$ is a T-dimensional vector whose *i*th element is $k(\mathbf{u}_i, \mathbf{x})$, and C is a *T*-by-*T* matrix whose *(m,n)* element is $k(\mathbf{u}_m, \mathbf{u}_n) + \beta^{-1}\delta_{mn}$. Also, $c = k(\mathbf{x}, \mathbf{x}) + \beta^{-1}$. $k(\cdot,\cdot)$ is called the kernel function, and in this study, we used a Gaussian kernel:

$$k(\mathbf{x}_m, \mathbf{x}_n) = \theta_1 \exp\left\{ -\frac{\theta_2}{2} \| \mathbf{x}_m - \mathbf{x}_n \|^2 \right\}.$$

The parameters θ_1, θ_2 and β are determined heuristically.

The GP-based forward model is expected to reproduce a trajectory similar to one of the exemplary trajectories D; this generalization capability is due to the smooth kernel function $k(\cdot,\cdot)$. It is also noted that in this imitation scheme, each trajectory to be imitated is realized by the target robot system rather than a human demonstrator. Then, each exemplary trajectory is consistent with the kinematics of the robot system.

2. Normalized Gaussian Network (NGnet)

The inverse model $p(P_{gt} \mid P_t, s_t, s_{t+1})$ is used to solve the inverse kinematics of the target robot system, that is, to output an appropriate control signal P_{gt}, given the current state s_t, the current internal state P_t, and the target next state s_{t+1}. The inverse model is represented by NGnet [3], a probabilistic model-based regressor. Although the inverse kinematics are often ill-posed, especially when the robot's DOF is large, this difficulty can be avoided by learning probabilistic relationship between the control signal and the consecutive robot states before and after the control signal is applied. Specifically, $p(P_{gt} \mid P_t, s_t, s_{t+1})$ is given by

$$p(P_{gt} \mid P_t, s_t, \hat{s}_{t+1}) = \sum_{k=1}^{M} w_k N(P_{gt} \mid W_K x_t + c_k, \Sigma_k) \quad (2)$$

where x_t denotes a concatenation of vectors P_t, s_t, and $\hat{s}_{t+1}$, and $w_k = \dfrac{g_k N(x_t \mid \mu_k, Q_k)}{\sum_{k'=1}^{M} g_{k'} N(x_t \mid \mu_{k'}, Q_{k'})}$.

During the free control phase (see below), the parameters $W_k, s_k, \Sigma_k, g_k, \mu_k$, and Q_k are estimated by variational Bayes (VB) estimation method [3]; though the VB method obtains the posterior distribution of the parameters, we just used their means in eq. (2). The NGnet divides the input space softly into sub-regions by means of Gaussian-based soft clustering, and linearly represents the relationship between the input and the output in each local sub-region, so that the final output is integrated such linear relationship like eq. (2). Since the relationship between the control input and the current and next states of the robotic arm should be nonlinear but is expected to be smooth within each local sub-region, the locally linear model like NGnet seems appropriate to well approximate such a moderate nonlinear relationship

IV. Experiment

1. Detailed experimental condition

1-1. Dimensionality reduction

Since the robotic arm system used in this study has 9 DOF and 26 McKibben actuators, a naïve control strategy could suffer from a curse of dimensionality. To reduce the redundant dimensions efficiently, we performed the dimensionality reduction in advance [4][5]. In particular, we used principal component analysis (PCA). Figure 2 shows the cumulative contribution rate against the number of PC dimensions when the robotic arm was randomly controlled. In this study, the state s_t is reduced to a four-dimensional vector s'_t, so that the cumulative contribution rate is 0.997.

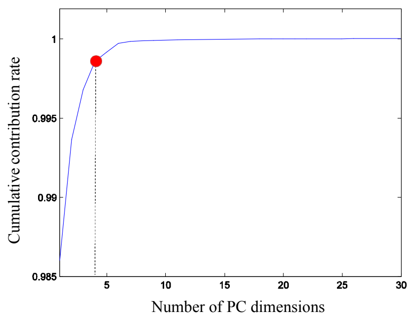

Fig.2. Dimensionality reduction by PCA

To avoid the extra complexity in controlling the robotic arm, only two McKibben actuators out of 26 McKibben actuators were controlled dynamically; the target air pressures of these two actuators were estimated by NGnet, and those of the rest of the McKibben actuators were fixed so that they maintain some stiffness. The two dynamically-controlled McKibben actuators were chosen heuristically after confirming they can lift the hand of the robotic arm up and down. The internal air pressure P_t is also reduced to such a two-dimensional vector that represents the internal air pressures of the two dynamically-controlled actuators.

1-2. Imitation phase

The exemplary trajectories were generated by moving the robotic arm manually by a human tutor. The tutor manipulated the robotic arm like handshake gesture. The exemplary trajectories consisted of 250 time points in total, including about 12 handshake motions, each of which had an approximately same period. Figure 3 shows a picture when the human tutor directly manipulated the robotic arm so that it imitates the handshake gesture.

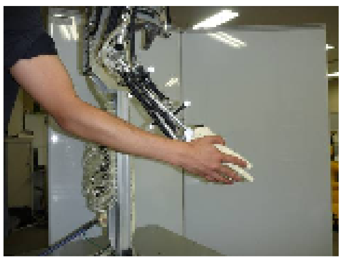

Fig.3. Human teaches the robot

1-3. Free control phase

In the free control phase, two periodic target air pressures were given to the two dynamically-controlled McKibben actuators. Each target air pressure was generated by a single sin function, but the periodicities of these sin functions were made to be different with each other so that they did not synchronize and the wide variety of the states was realized. 40,000 data points were collected at the sampling frequency of 200Hz, and used to train the inverse model (NGnet).

To determine the number of components of the NGnet, M in eq. (2), we performed "Ten-fold cross validation" [6]. Figure 4 shows the result. Since the cross-validation error decreases as M increases for these M values, we set $M = 30$; note that the computation cost becomes too much if the number of components is larger than 30.

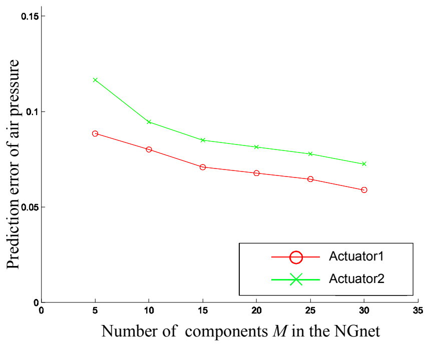

Fig.4. Ten-fold cross-validation

Figure 5 depicts the information flow of our contr ol system.

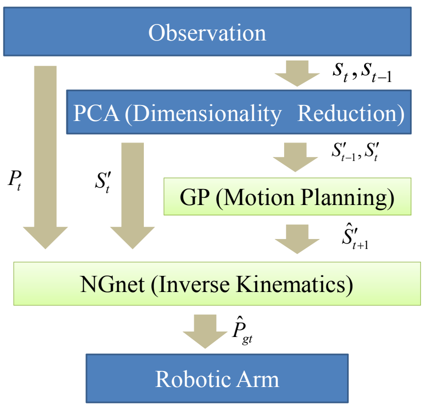

Fig.5. Information flow in our scheme

2. Results of the experiment

Figures 6 and 7 show the trajectories of the first and the second principal components of the states, respectively, when the trained robotic arm reproduces the handshake motions in the real-time control. The blue line is an exemplary trajectory, and the red and green lines are the trajectories realized by our control system before and after the training, respectively.

As can be seen in the figure, the robotic arm successfully reproduced a smooth and periodic trajectory after the training. In accordance with the figure, we visually confirmed the robotic arm performed a natural handshake gesture.

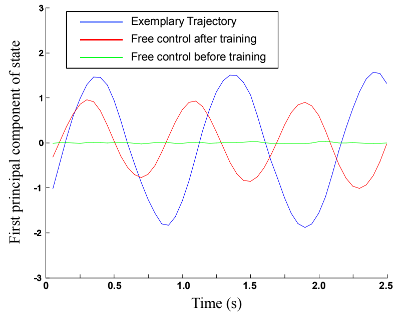

Fig.6. Trajectory of the first PC

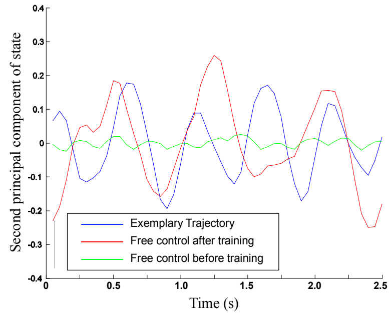

Fig. 7 . Trajectory of the second PC

Ⅶ. Conclusion

In this study, we propose to employ machine learning-based regressors to perform a real-time motion planning and to solve the inverse kinematics of the robot. The real-time motion planning (forward model) enabled the robotic arm to reproduce a natural handshake gesture as the human tutor taught and the learning of the inverse kinematics (inverse model) enabled real-time computing of the control input. It is expected that the proposed framework is applicable to other robots, and to perform other motions, since our framework is not task-specific but general enough.

Reference

[1] Noritsugu T (1997), Pneumatic Actuators (in Japanese). J. Robot. Soc. Japan, 15:355-359
[2] Rasumssen CE, Williams CKI (2006), Gaussian Processes for Machine Learning, MIT Press
[3] Yoshimoto J, Ishii S, Sato M (2003), Normalized Gaussian Network Based on Variational Bayes Inference and Hierarchical Model Selection (in Japanese), SICE J. Cont., Meas., and Syst. Int., 39(5):503-512
[4] Bishop CM (2006), Pattern Recognition and Machine Learning, Springer
[5] Calinon S, Guenter F, Billard A (1995), On Learning, Representing and Generalizing a Task in a Humanoid Robot, Complexity, 1:12-1
[6] Kohavi R (1995), A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. in Proc. Int. Joint Conf. Artificial Intelligence, 2:1137-1143