2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)
Mexico City, Mexico. September 11-13, 2019

Human Behavior Learning in Joint Space Using Dynamic Time Warping and Neural Networks

Jorge Ramírez
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico
Email: jramirez@ctrl.cinvestav.mx

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN
Mexico City, Mexico
Email: yuw@ctrl.cinvestav.mx

***Abstract*—The usual human behavior learning methods are in the robot task space, i.e., three-dimension Cartesian space. After learning, the desired trajectories have to be transformed into the joint space by the inverse kinematics of the robot. However, for most robots, the analytical solutions of the inverse kinematics cannot be obtained.**

In this paper, we learn human behavior directly in the joint space. There are some problems to learn the demonstrations in the joint space, such as the demonstration trajectories depending on different velocities and tremors in Cartesian space produced by natural human behavior. We use dynamic time warping and neural networks to solve these problems. More importantly, we avoid calculating the inverse kinematics. Experiment results have shown this method to be effective.

I. Introduction

The use of robots to assist and perform a wide variety of tasks has increased in recent years, the existing technology is capable of performing tasks with very high precision and speed, far exceeding this way humans who try to perform the same tasks. However, these robots are controlled manually or programmed only to follow a predesigned specific path. The development of intelligent robots capable of performing selective autonomous movements could improve and increase the capacities to develop a particular task.

Demonstrations are based on the natural human behavior since the movements are skills the human performs as useful trajectories that the robot can learn [1] [2]. The human movement occurs through movements in the articulations, which describes a trajectory in the joint space. This behavior is translated as an action that is found in the task space, as it could be drawing a figure or moving an object from one point to another. The classical methods [3] allows to calculate the necessary joint angles to perform actions in the task space, but these present problems, since it is mathematically difficult to define a single optimal trajectory or it is not possible to obtain an analytical solution (as it is in the presence of redundancy). Learning a useful trajectory generated by a human expert ensures that the task will be performed without mathematical impediments of kinematics.

978-1-7281-4840-3/19/$31.00 ©2019 IEEE

The demonstrations are made by a subject called teacher, the subject who learns the demonstration is called an apprentice. This technique of learning human behavior is divided into two phases: behavior learning and behavior generation; the first one refers to how the demonstrations are mapped to the robot and the second refers to the reproduction of the desired behavior trajectories by the controller [4].

Learning from Demonstration (LfD) is a technique that allows mapping between states and actions, being able the robot to generate an action based on demonstrations, where the states and actions are stored and learned using different techniques that allow generating the original or similar actions. In [5] a further explanation is given about the LfD technique.

The way how demonstrations based on the action-state relationship are carried out is essential for obtaining data since this depends on the robot complexity and the task that it is intended to teach.

This paper presents a method of learning tasks based on demonstrations made by humans, learning to perform the task independently with improved performance in speed and accuracy. Similar methods in the joint space has been developed before [6], [7] and [8].

II. Human behavior learning

In most LfD, a human being is used as a teacher to perform the demonstrations, although a robot or a planning simulation can also be a teacher. In the election of the teacher, it is necessary to consider (i) who will control the demonstration and (ii) who will execute the demonstration. The strategy of how the data learner is provided is divided into two, (i) Batch learning, where learning is done once all the data has been obtained and (ii) iterative learning, where learning is performed as the data is available.

Correspondence is a significant factor to take into account since the apprentice, and the teacher can be morphologically different, the teacher having greater or lesser degrees of freedom than the apprentice, different ranges of movement or links between joints of different sizes. It is, therefore, necessary to take into account only some states or make a mapping to match them. The correspondence can be categorized into:

- Record Mapping: it refers to the coincidence between the states-actions experienced by the teacher and those stored.
- Embodiment Mapping: it refers to the coincidence between the states-actions stored and those executed by the apprentice.

The data obtention can be separated into two categories based on the demonstration execution platform and its correspondence in (i) Demonstration and (ii) Imitation.

It is considered as a demonstration when the teacher uses the apprentice to execute the task. There are no correspondence problems due to embodiment mapping when the apprentice is used because the action-states experienced by the apprentice sensors are directly stored. A demonstration can be categorized into the next two techniques:

- Teleoperation: is the demonstration technique where the teacher operates directly over the apprentice who through its sensors stores the states-actions experienced by the teacher. No embodiment or record mapping is necessary.
- Shadowing: is the demonstration technique where the apprentice tries to follow the movement of the demonstration executed by the master, the apprentice uses its sensors to store the action-states. Record mapping is necessary.

It is considered an imitation when the sensors are mounted outside the apprentice's platform. Then the stored action-states will be different from those executed by the apprentice. Therefore an embodiment mapping will be necessary. LfD based on imitation can be classified into:

- Sensors in the master: is an imitation technique where the sensors are mounted directly on the master's execution platform. No record mapping is necessary.
- External observation: this technique of imitation occurs when the sensors are outside the operating platform of the master and the apprentice, a record mapping is necessary.

There are methods which can be variations of LfD but cannot be categorized within the previous sections, because in most cases, only the states are stored and not the actions.

The learning techniques for the stored action-state can be categorized in:

- Mapping function: mapping function tries to approximate by a function, from the states to the action. The objective of this technique is to reproduce the teacher fundamental behavior and generalize about the stored action-states, then find a solution for a set of states similar to the demonstration ones.
- System model: the system model technique makes use of a state transition model $T(e'|e,a)$ which is determined through the stored action-states, with the help of a reward function $R(e)$ that can be learned by demonstrations or designed by the user, the value of the reward function r is associated with the state e that the apprentice will reproduce.
- Planning: this LfD technique is based on generating the desired behavior of the learner as a plan, thus being a sequence of actions ranging from the initial state to the final state.

III. Demonstrations processing in joint space

The LfD states stored are given by a signal in discrete-time, done by a sampling process that replaces the continuous-time with a series of values in the discrete-time thus the measurements taken by the servomotors sensors are given intermittently.

Demonstrations are sets of discrete-time joint signals represented by a sequence of N numbers denoted as $q_x^y(n)$, where $n = 1, 2, 3, ..., N$, x is the joint number and y is the demonstration number. The complete set of ϕ joint signal corresponding to the y demonstration is expressed as

$$q^y = \begin{bmatrix} q_1^y(n) \\ q_2^y(n) \\ \vdots \\ q_\phi^y(n) \end{bmatrix} \tag{1}$$

A. Dynamic time warping for demonstrations alignment

It is known as Dynamic Time Warping (DTW) the algorithm capable of analyzing and measuring similarities in time series that can be different in speed [9], in the same way, an optimal alignment can be found between these time series to be deformed. In [10], DTW is used to measure similarities in music melodies.

To align two different demonstrations, first, it starts from any individual joint x

$$q_x^1(n) = \{q_x^1(1), q_x^1(2), q_x^1(3), ..., q_x^1(i), ..., q_x^1(N)\} \tag{2}$$

$$q_x^2(m) = \{q_x^2(1), q_x^2(2), q_x^2(3), ..., q_x^2(j), ..., q_x^2(M)\} \tag{3}$$

In order to align these joint signals using DTW, it is necessary to form a matrix where the Euclidean distance defines each element as

$$d_{ij}^x = |q_x^1(i) - q_x^2(j)| \tag{4}$$

Subsequently, the deformation path for this joint is denoted by

$$W^x = w_1^x, ..., w_{k-1}^x, w_k^x, ..., w_{l_x}^x \tag{5}$$

where $w_1^x = (1,1)$, $w_{k-1}^x = (a_{k-1}^x, b_{k-1}^x)$, $w_k^x = (a_k^x, b_k^x)$ and $l_x \leq N + M - 1$, where a_k^x and b_k^x are the subscripts of the aligned joint signals which are given by

$$\begin{aligned} r^x(a_k^x, b_k^x) = \min\{&d^x_{(1+a^x_{(k-1)}),(1+b^x_{(k-1)})}, \\ &d^x_{(a^x_{(k-1)}),(1+b^x_{(k-1)})}, d^x_{(1+a^x_{(k-1)}),(b^x_{(k-1)})}\} \end{aligned} \tag{6}$$

W^x is different for every joint signal. It is necessary to apply the same deformation path to the whole set of articulations to not alter the actions of each demonstration in the task space when aligning each articulation.

The average deformation path of alignment for ϕ number of joints is defined as follows

$$l_{min} = \min\{l_1, l_2, l_3, ..., l_\phi\} \tag{7}$$

$$\begin{aligned} w_1^* &= \frac{1}{\phi}(w_1^1 + w_1^2 + w_1^3 + \ldots + w_1^\phi) \\ w_2^* &= \frac{1}{\phi}(w_2^1 + w_2^2 + w_2^3 + \ldots + w_2^\phi) \\ &\vdots \\ w_{l_{min}}^* &= \frac{1}{\phi}(w_{l_{min}}^1 + w_{l_{min}}^2 + w_{l_{min}}^3 + \ldots + w_{l_{min}}^\phi) \end{aligned} \tag{8}$$

$$W^* = w_1^*, w_2^*, ..., w_{l_{min}}^* \tag{9}$$

B. Change of velocity

If a_k^* and b_k^* are chosen to form a deformation path where $q_x^1(n)$ meets the same trajectory as $q_x^2(n)$ at a different velocity the following algorithm can be deduced, given any joint signal

$$q_x^y(n) = \{q_x^y(1), q_x^y(2), q_x^y(3), ..., q_x^y(i), ..., q_x^y(N)\} \tag{10}$$

the step of intermediate sampling data is defined based on the desired speed

$$j = round\left(\frac{1}{vel}\right) \tag{11}$$

where the desired velocity $vel \in (0, 1]$.

Then, the subscripts of the new time series are

$$\begin{aligned} k_1 &= j \\ k_2 &= k_1 + j \\ &\vdots \\ k_{\tilde{N}} &= k_{\tilde{N}-1} + j \end{aligned} \tag{12}$$

where $\tilde{N} = round\left(\frac{N}{j}\right)$.

So that

$$q_x^{vel}(k_{\tilde{n}}) = \{q_x^{vel}(k_1), q_x^{vel}(k_2), ..., q_x^{vel}(k_{\tilde{N}})\} \tag{13}$$

IV. Human behavior learning using neural network

A neural network is a model that attempts to simulate a fundamental information processing method of a biological brain [11], and this can be used as a learning technique for LfD. The neural network abstracts the provided demonstrations, generating a knowledge that can later be used to generate trajectories that reproduce a natural human behavior. Human behavior skills as been modeled using neural networks in [12].

A. Neural network

Feedforward neural networks allow input signals to go only in one direction, from input to output. In this architecture, there are no cycles or backward connections.

Huang and Brabri showed [13] that a feedforward neural network with a single hidden layer with at most κ number of hidden neurons with almost any non-linear activation function could learn exactly κ different observations. Based on this, a feedforward neural network with a single hidden layer is proposed for a κ number of neurons, the relationship between inputs and synaptic weights can be expressed in matrix form as

$$u = A^T x \tag{14}$$

and

$$v = G(u) \tag{15}$$

where the number of inputs is $\psi = 3(p+1)$, $G(\cdot)$ is the respective activation function and p is the dynamic memory which helps to system modeling [11] and

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1\kappa} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2\kappa} \\ \vdots & \vdots & & \vdots \\ \alpha_{\psi 1} & \alpha_{\psi 2} & \ldots & \alpha_{\psi\kappa} \end{bmatrix} \quad x = \begin{bmatrix} x(n) \\ x(n+1) \\ x(n+2) \\ \vdots \\ x(n+p) \\ y(n) \\ y(n+1) \\ y(n+2) \\ \vdots \\ y(n+p) \\ z(n) \\ z(n+1) \\ z(n+2) \\ \vdots \\ z(n+p) \end{bmatrix} \tag{16}$$

For the output layer with ϕ number of outputs, it can also be written in matrix form as

$$\hat{q} = Bv \tag{17}$$

where

$$B = \begin{bmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1\kappa} \\ \beta_{21} & \beta_{22} & \ldots & \beta_{2\kappa} \\ \vdots & \vdots & & \vdots \\ \beta_{\phi 1} & \beta_{\phi 2} & \ldots & \beta_{\phi\kappa} \end{bmatrix} \quad \hat{q} = \begin{bmatrix} \hat{q}_1(n) \\ \hat{q}_2(n) \\ \vdots \\ \hat{q}_\phi(n) \end{bmatrix} \tag{18}$$

The proposed neural network is trained with a simple and very efficient algorithm called Extreme Learning Machine (ELM) presented in [14] and [15], which is used to train feedforward neural network with a single hidden layer, where random weights $\alpha_{\psi\kappa}$ are assigned, the weights connected to the hidden layer $\beta_{\phi\kappa}$ are obtained trough the generalized Moore-Penrose inverse of the matrix v.

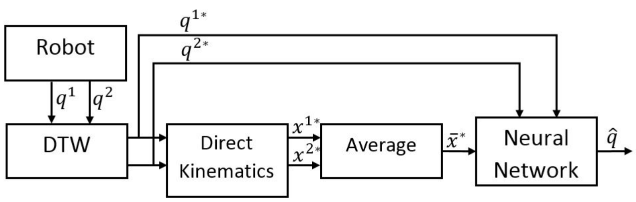
B. Learning error

To evaluate the performance of the inverse kinematics process learned by the neural network, a trajectory different from the used for training as a test is used and compared with the direct kinematics of the neural network output signals.

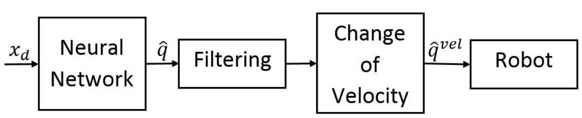
The inverse kinematic error is calculated through the mean square error expressed as

$$MSE = \frac{1}{3N}\sum_{n=1}^{N}((x_d(n) - \hat{x}(n))^2 + (y_d(n) - \hat{y}(n))^2 + (z_d(n) - \hat{z}(n))^2) \quad (19)$$

where $\hat{x}(n)$, $\hat{y}(n)$ and $\hat{z}(n)$ are the direct kinematics spatial coordinates of the neural network output signals and $x_d(n)$, $y_d(n)$ and $z_d(n)$ are the desired trajectory to reproduce the human behavior in task space.



(a) Behavior learning phase



(b) Behavior generation phase

Fig. 1: Learning from demonstration

V. Experimental Results

For the development and experimentation is used the Cyton Gamma 1500 manipulator robot [16], built and designed by Robai Corporation, which is a redundant humanoid robotic arm, in the same way to a human arm, thus allowing to position and orient the robot in a significant number of ways. It has seven degrees of freedom; each joint and the gripper are controlled and moved by individual servomotors.

A human being was used to directly manipulate the apprentice robot body to trace desired trajectories in the robot task space, being the human being who controls the demonstration and the apprentice robot who executes the demonstration.

An iterative type demonstration technique was used.

In order to obtain data, a demonstration technique was used, with the apprentice as the demonstration platform using the sensors mounted on the apprentice body, record and embodiment mapping were not necessary. The demonstration technique used is considered within the exceptions because only the robot states were stored, and the direct kinematics of the robot generated the action.

As a learning technique, is used a regression-type function mapping implemented through a feedforward neural network with a single hidden layer and trained through the ELM algorithm.

In LfD where the human factor is involved and is desired to increase the generality of learning, it is difficult to reproduce the same trajectory at the same speed in each performed demonstration, since the human being has an error of position and velocity trying to repeat the same action.

For this case of study, it is proposed to train the neural network using two similar demonstrations that reproduce the human behavior of figure drawing that represents an eight on a plane, drawn at different velocities. The states of the demonstrations obtained through the servomotor sensors in each of the seven articulation of the robot are shown in the figure 2 where the demonstration 1 lasts 5.25 seconds and the demonstration 2 lasts 2.97 seconds, the actions generated by the direct kinematics of the states are shown in the figure 3.

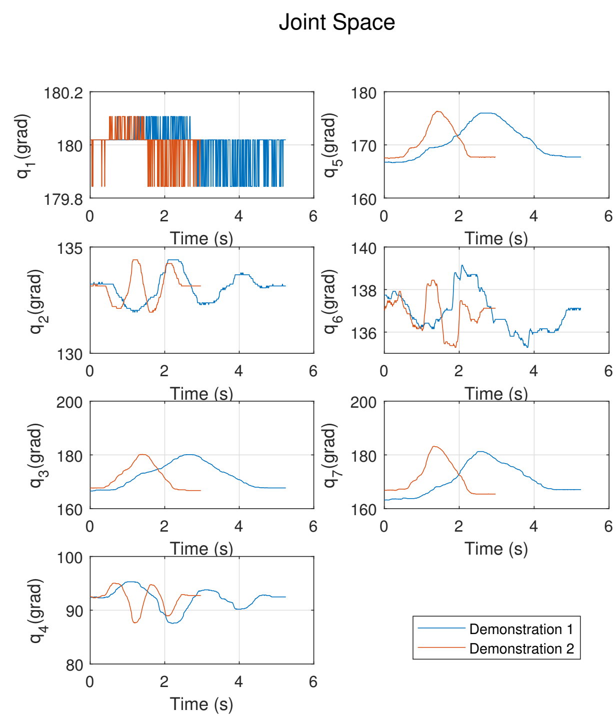


Fig. 2: Demonstrations states

Following the diagram in figure 1a to teach these demonstrations, it is necessary to make a general alignment of the states using the DTW algorithm developed in section III-A, so that the deformation path applied for the alignment is the same for each state, the action represented by the states cannot be affected.

The aligned demonstrations are used to train the proposed feedforward neural network architecture in section IV-A, with a number of neurons $\kappa = 700$ on his hidden layer and a dynamic memory of $p = 2$. Therefore the number of inputs is $\psi = 9$. The neural network input signal is the action

Fig. 3: Direct kinematics of the demonstrations

generated by the arithmetic mean of the direct kinematics of the aligned demonstrations. As there is no need for record and embodiment mapping, the number of neural network outputs is equal to the number of states in the demonstrations $\phi = 7$.

Once the neural network has been trained, it is possible to generate human behavior according to the diagram in figure 1b. The neural network outputs generates signals with high frequencies corresponding to hard to learn trajectories and demonstrations containing typical human tremors. It is desired to keep the frequencies lower than 3.5 Hz and attenuate the higher ones, for which a low pass Butterworth digital filter is designed of order 2, given by the transfer function

$$H(z) = \frac{0.0104(z+1)(z+1)}{(s-0.8455-0.1336i)(s-0.8455+0.1336i)} \tag{20}$$

The digital filter applied to the states in the output of the neural network is shown in figure 4 and his direct kinematics are compared with the reference trajectory seen in figure 5, wherein the resulting trajectory does not contain the vibrations produced by the teacher's muscular tremors.

In figure 6 it is observed that when accelerating 10 times the speed of the filtered neural network outputs, the direct kinematics trajectory remains similar to the original one (6a), while when accelerating 20 times the velocity (6b) it presents small data losses that impact on small deformations in the direct kinematics trajectory, the loss of data is substantial when accelerating 30 and 40 times (6c and 6d), impacting on a significant deformation of the original direct kinematics trajectory.

VI. Conclusion

In this paper, we have presented a method capable of teaching to a robot a task that reproduces the natural human behavior, showing in this way that the method improves in speed and precision the performance of a human. Since the learning is in the joint space, the main advantage over the other human behavior learning methods is that our method

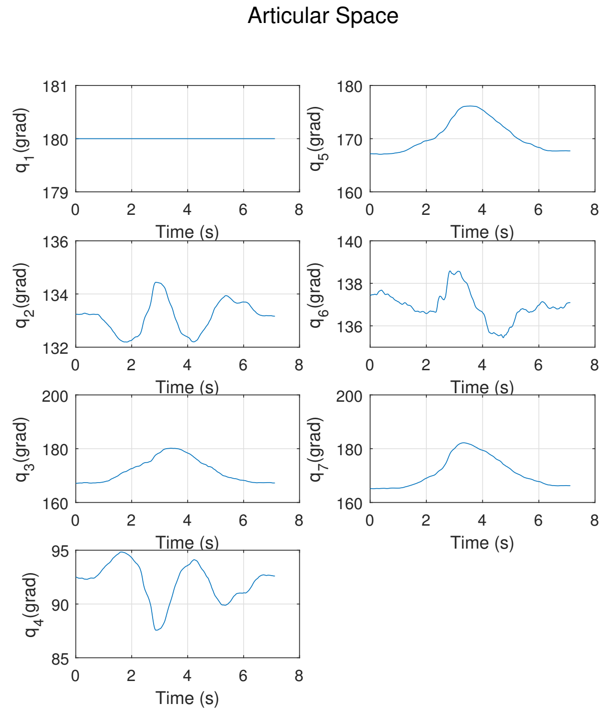


Fig. 4: Filtered neural network outputs

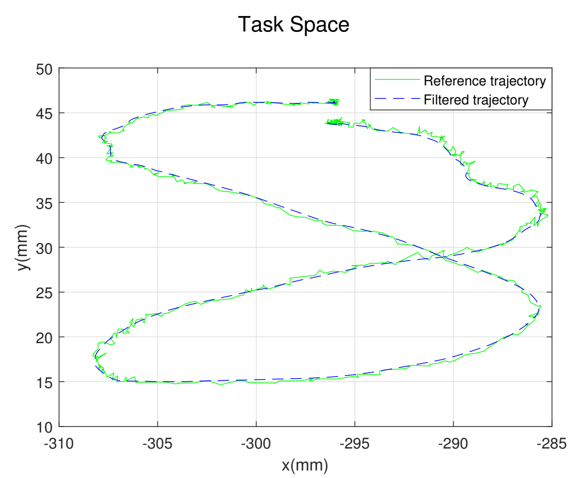
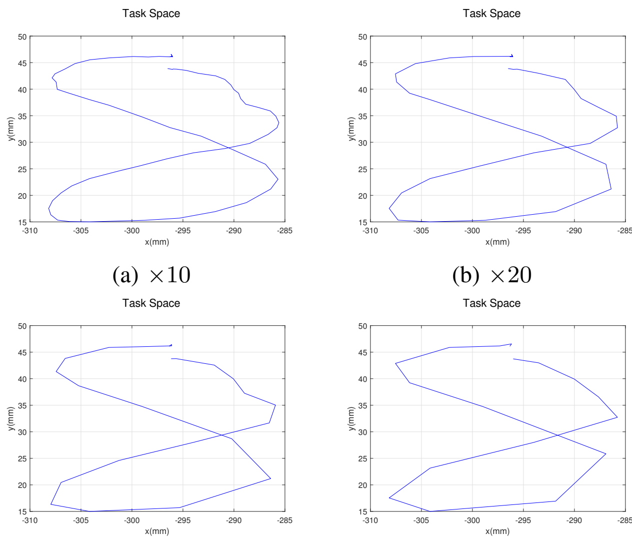


Fig. 5: Comparison of the reference trajectory and the direct kinematics of the filtered neural network outputs

does not need the inverse kinematics of the robot. The use of current algorithms that improve the speed of learning in neural networks has facilitated the development of this method. However, the learning can be severely affected, given the complexity of the task to be learned, because of the more complex the task is, the higher the number of neurons needed in the hidden layer.



(a) $\times 10$ (b) $\times 20$

(c) $\times 30$ (d) $\times 40$

Fig. 6: Direct kinematics of the neural network output with a change of velocity

Acknowledgment

The first author would like to thank CONACYT and CINVESTAV for the great opportunity given, all the support and facilities.

References

[1] Y. Xu and K. K. C. Lee, *Human Behavior Learning and Transfer*. CRC Press, 2005.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaa, *Robot Programming by Demonstration*. Springer, 2008.

[3] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*. John Wiley and Sons Inc., 1989.

[4] J. G. Melendez, "Aprendizaje por demostración en el espació articular para el seguimiento de trayectorias aplicado en un exoesqueleto de 4 grados de libertad," Ph.D. dissertation, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, 2015.

[5] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems, 469-483*, 2009.

[6] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning fromhuman-guided demonstration," *IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA pages 2074-208*, 2010.

[7] J. Peters and S. Schaal, "Learning to control inoperational space," *The International Journal of Robotics Research*, 2008.

[8] J. Garrido, W. Yu, and A. Soria, "Human behavior learning for robot in joint space," *Elsevier*, 2015.

[9] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," *AAAI Technical Report WS-94-03*, 1994.

[10] H. Yu and I. Henríquez, "Music classification based on melodic similarity with dynamic time warping," *IEEE International Conference on Computational Intelligence and Computing Research*, 2013.

[11] S. Haykin, *Neural Networks a Comprehensive Foundation*. Prentice-Hall, 1999.

[12] H. Asada and S. Liu., "Transfer of human skills to neural net robotcontrollers," *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991.

[13] G.-B. Huang and H. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Trans. Neural Networks*, 9(1)(1998)224-229.

[14] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *ELSEVIER*, 2006.

[15] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *ELSEVIER*, 2015.

[16] *ROBAI. Cyton Gamma 1500 Arm Specifications*.

[17] K. Ogata, *Discrete-Time Control Systems*. Pearson, 1995.

[18] F. W. Isen, *DSP for MATLAB and LabView Volume III: DigitalFilter Design*. Morgan and Claypool, 2009.

[19] J. M. Giron-Sierra, *Digital Signal Processing with Matlab Examples*. Springer, 2017.

[20] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Pearson, 2010.