Hindawi Publishing Corporation
Journal of Robotics
Volume 2012, Article ID 505191, 15 pages
doi:10.1155/2012/505191

Research Article

Robots Learn Writing

Huan Tan,[1] Qian Du,[2] and Na Wu[3]

[1] *Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37240, USA*
[2] *Institute of Robotics and Automatic Information System, Nankai University, Tianjin 300071, China*
[3] *Graduate School of Decision and Technology, Tokyo Institute of Technology, Tokyo 152-8552, Japan*

Correspondence should be addressed to Huan Tan, huantan@ieee.org

Received 19 March 2012; Accepted 19 June 2012

Academic Editor: Huosheng Hu

This paper proposes a general method for robots to learn motions and corresponding semantic knowledge simultaneously. A modified ISOMAP algorithm is used to convert the sampled 6D vectors of joint angles into 2D trajectories, and the required movements for writing numbers are learned from this modified ISOMAP-based model. Using this algorithm, the knowledge models are established. Learned motion and knowledge models are stored in a 2D latent space. Gaussian Process (GP) method is used to model and represent these models. Practical experiments are carried out on a humanoid robot, named ISAC, to learn the semantic representations of numbers and the movements of writing numbers through imitation and to verify the effectiveness of this framework. This framework is applied into training a humanoid robot, named ISAC. At the learning stage, ISAC not only learns the dynamics of the movement required to write the numbers, but also learns the semantic meaning of the numbers which are related to the writing movements from the same data set. Given speech commands, ISAC recognizes the words and generated corresponding motion trajectories to write the numbers. This imitation learning method is implemented on a cognitive architecture to provide robust cognitive information processing.

1. Introduction

Robots are expected to generate human-like behaviors in dynamic environments [1, 2]. However, it is very difficult for robots to develop skills or behaviors totally from scratch without any initial knowledge. As stated in Sloman's paper, robots should learn both altricial and precocial behaviors after their "birth" [3]. Therefore, it is reasonable that robots have some basic and simple initial knowledge with motion primitives [4], or some basic and simple initial skills to explore the world to develop new knowledge and skills to survive or complete tasks. Upon these initial knowledge and skills, humans can teach robots more complex behaviors or skills to complete much more complex tasks.

Imitation learning (also called learning from demonstration, programming by demonstration) is now considered as a powerful tool for transferring skills between robots (especially humanoid robots) [5]. Unlike the traditional teaching-executing mode, where robots simply record the trajectory programmed by human operators and move the angles and end effector along the trajectory, since 1970s, the researchers had tried to train robots to learn simple motion patterns [6]. In 1980s, Atkeson trained a robot to learn how to balance an inverted pendulum in an upright position through practice [7]. From then, many methods in imitation learning have been proposed in various areas [8]. In 2000s, researchers found biological evidence and models of imitation learning in animals [9]. Gradually, imitation learning has been divided into two parts [10]. one is to train robots to learn the dynamics of movements [11] and the other is to train robots to learn the primitives in a behavior sequence [12].

The motivation of this paper is to find a method that robots can learn motion models and semantic knowledge simultaneously in the current popular imitation learning framework. In the experiment part, a humanoid robot, named ISAC, is trained to learn writing numbers from a human teacher.

The rest of this paper is organized as follows. Section 2 introduces the current related work; Section 3 explains the system framework and the algorithms used in this framework; Section 4 explains the implementation on a cognitive architecture; Section 5 explains the experiment setup and the experimental results; Section 6 discusses the experimental results and the future work; Section 7 concludes the work in this paper.

2. Related Work and Motivation

2.1. Motion Learning. Demonstrations of motions are given by human teachers or other robots and a robot student tries to record the demonstrations. There are many different kinds of methods for demonstrating the motions: learning from observation [13], from joystick operation [14], by manually moving the arm of a robot [10], and from the sensor on the human body [15, 16].

Sometimes the dimension of the recorded data is reduced by projecting the data from a high-dimensional data space to a low-dimensional data space, named latent space. Correspondingly, the data needs to be reconstructed from the low-dimensional data space to the high-dimensional data space. The "dimension reduction" and the "reconstruction" are not always required in current imitation learning research. In some situations where the dynamics of the demonstrations need to be analyzed or several inner correlations need to be analyzed, the "Dimension Reduction" and the "Reconstruction" can be applied. Many dimension reduction methods are proposed to extract the features of the data such as principal component analysis [17], factor analysis [18], ISOMAP [19], local linear embedding [20], and MDS [21]. A typical example of using dimension reduction technology is [10], in which Calinon and Billard proposed a method to utilize the dimension reduction methods to establish a strong coupling relationship between the data in the latent space and the data space, and use the data distribution in the latent space to ensure the generated behavior has similar inner dynamics and constraints to the demonstrations.

The learned motion models are stored in the memory (database) of robots, where robots store the learned knowledge or skills. Linear Global Model (LGM) [22], Gaussian Process (GP) [23], Locally Weighted Regression (LWR) [24], Locally Weighted Projection Regression (LWPR) [25], Principal Curve (PC), Gaussian mixture model [26], and Artificial Neural Network (ANN) [27] are used for representing the models in the memory.

2.2. Semantic Knowledge Learning. Robots need to understand the learned motions, and it means that robots need to relate these motion models to corresponding semantic knowledge. This is normally done by labeling the motion models with a semantic name or with a semantic description of related tasks.

2.3. Generation. Given a similar and slightly different situation (where robots need to complete a same type of task with different parameters), a command, or an outside trigger (signal, image, etc.), required actions are planned and required motion models are retrieved by searching their corresponding behavior names in the "labeled behavior models."

If needed, the parameters of the motions are modified to adapt to the similar but slightly different situation. The generated behaviors are described as actions with specified parameters. Dynamic Movement Primitives (DMPs) [16] are widely for generating motions which have similar dynamics to the demonstrations and can achieve various targets. Calinon et al. proposed a method to minimize the weighted distance between the generated motions and the learned motions both in the latent space and in the original data space [10]. Peters used reinforcement learning [28] methods for robots to adapt the parameters of motion models to generate similar motions in similar but slightly different situations [29]. Theodorou applied optimal control [30] in reinforcement learning environment for robots to learn the motion models of demonstrations and generate similar motions using DMP [31, 32].

If the data is stored in a latent space, the generated trajectories of motions need to be projected from the latent space to the original data space, for example, joint space.

2.4. Motivation. Robots need to learn the motions of the behaviors and the semantic meaning of the behaviors at two stages as described in former sections. One of the learning stages is still like a programming process, in which the behavior names are assigned to the motion models manually by human teachers.

An important problem is that robots can learn the motions of the demonstrations and use the learned motions in a similar but slightly different situation, but how can robots use such learned knowledge in other area besides executing movements, for example, recognition, semantic understanding, reasoning, and planning, Especially in writing, the learned movements should relate to semantic meanings of the letters, numbers or symbols, and robots may use the learned movements to find their higher level semantic meanings. When we see someone demonstrate how to a character, we can have direct thought about the meaning of that character in our brain, and when we have the method of writing a character in our brain, we can evaluate the results with a real character. A game which may be familiar with most of us is that someone writes letters on our back using fingers and we try to guess what he/she is writing. Obviously, we, humans, can use the sensing information on the back to construct the trajectory of the movement of the finger in our brain, and try to compare it with our learned knowledge about the letters. In this paper, the robot uses encoders to sense the movement of the joints in the joint space and tries to match the sensed movement of the joints with the learned knowledge about the numbers. Then, human teachers do not need to teach the robot that what the number is and the robot can automatically relate the learned movements of writing letters to the corresponding letters.

3. System Design

In this paper, we proposed a general framework, using which robots can complete two learning stages mentioned above simultaneously, which is shown in Figure 1.

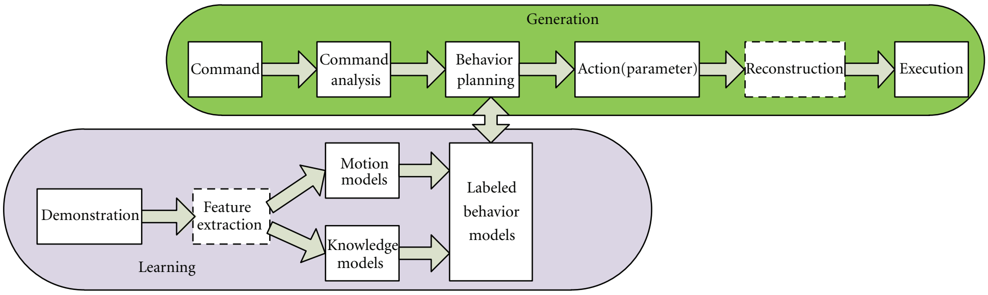


Figure 1: Proposed framework for imitation learning.

From Figure 1, the contribution of this proposed framework is that the information for both motion models and the knowledge models is both from one single source. The robot uses the information of the demonstrations to learn motion models and the knowledge models. In this paper, ISAC learn how to write numbers and to automatically relate the motions of writing numbers to their semantic knowledge models.

3.1. *Demonstration.*

3.1. Demonstration. Demonstrations are given by human teachers. In this paper, demonstrations are shown by manually moving the right arm of ISAC.

The recorded data is $\theta_s = \{\theta_v, t\}$, which is an $N \times 7$ matrix. θ_v records the angles of six joints on the right arm of ISAC, and t is the temporal information.

3.2. Feature Extraction. For most situations, robots need to learn two features: motions and semantic meaning. The extracted information is stored in corresponding models.

3.2.1. Motion Models Learning. As mentioned in the introduction section, there are many methods to represent the data of motions. In this paper, we use a modified ISOMAP algorithm [33] to project the 6-dimensional data space to a 2-dimensional space. The movements of the writings are in a 3-dimensional Cartesian space which is driven by 6 joints. However, the features of the writings are 2-dimensional because the characters are written on a 2-dimensional plane. So it is reasonable to use the features on a 2-dimensional plane for other use. The additional motivations of using this modified ISOMAP algorithm is to visualize the sampled data on 2-dimensional plane for researchers to find the features of motions easily, and to make the trajectory on the 2-dimensional plane does not have overlapped parts inside itself or have intersections inside itself. Using this algorithm, the spatial and temporal characteristic of the sample data can be visualized on a 2-dimensional plane.

We want to mention that dimension reduction is not necessary for all applications. In this paper, the dimension reduction is convenient for extracting the features and using the features in recognition.

The algorithm of the original ISOMAP algorithm is specified as follows.

(1) Sample the points on the demonstration trajectory:

$$\theta_s = \{\theta_v, t\}. \tag{1}$$

(2) Compute the geodesic distance matrix D_{Ms}:

$$\begin{gathered} D_{\mathrm{Gs}}(i,j) = \begin{cases} \left\|\theta_{vi} - \theta_{vj}\right\|, & \text{if } \left\|\theta_{vi} - \theta_{vj}\right\| \leq d \\ 0, & \text{otherwise} \end{cases} \\ D_{\mathrm{Ms}}(i,j) = \min\left(D_{\mathrm{Gs}}(i,j), D_{\mathrm{Gs}}(i,k) + D_{\mathrm{Gs}}(k,j)\right), \\ k = 1, 2, \ldots, N, \end{gathered} \tag{2}$$

D_{Ms} is iteratively calculated until the values of the elements are converged.

In the original ISOMAP algorithm, $\|\theta_{vi} - \theta_{vj}\|$ is defined as an Euclidean spatial distance between two points: θ_{vi} and θ_{vj}. In our modified ISOMAP algorithm, $\|\theta_{vi} - \theta_{vj}\|$ is defined as a temporal distance between two points: θ_{vi} and θ_{vj}. θ_v is used to record the angles of the six joints: $\theta_v = \theta_{v1}, \theta_{v2}, \ldots, \theta_{vN}$.

(3) Compute the inner products,

$$\tau(D_{\mathrm{Ms}}) = -\frac{1}{2}HSH, \tag{3}$$

where $S_{ij} = D_{\mathrm{Ms}ij}{}^2$, and $H = \delta_{ij} - (1/N)(\delta_{ij} = 1$, when $i = j$, $\delta_{ij} = 0$, when $i \neq j)$.

(4) Compute the new coordinates of the sampled points in the latent space X:

$$x_i = \left(\sqrt{\lambda_{s1}}\alpha_{s1}, \sqrt{\lambda_{s2}}\alpha_{s2}\right)^T, \tag{4}$$

where λ_1 and λ_2 are the two largest eigenvalues of $\tau(D_M)$ with two corresponding eigenvectors: α_1 and α_2.

The modified ISOMAP method, which reflects both the temporal and spatial relationship between sampled data points on a two-dimensional plane, is used to train robots to learn the motions of writing the letters on a two-Dimensional plane [33].

The original ISOMAP is an extension of the MDS, which constructs the distance matrix by connecting the sampled points through the neighbors. The original ISOMAP is used to describe the distance between the sampled neighbor points. In order to find the temporal information of the sampled trajectory, in our algorithm the neighbors are strictly defined as temporal neighbors. The spatial relationships are not defined but calculated by this modified ISOMAP algorithm. The modification is designed in

$$D_{\mathrm{Gt}}(i,j)=\begin{cases}\left\|\theta_{vi}-\theta_{vj}\right\|, & \text{if } \left|t_i-t_j\right|\le s,\\ 0, & \text{otherwise,}\end{cases} \quad (5)$$

where s is the temporal threshold value. In (1), d is the spatial threshold value.

Using this method, D_{Gt} and the corresponding $\tau(D_{\mathrm{Mt}})$ are calculated. The sampled points in the latent space are represented as $y_i=(\sqrt{\lambda_{t1}}\alpha_{t1},\sqrt{\lambda_{t2}}\alpha_{t2})^T$, where λ_{t1} and λ_{t2} are the two largest eigenvalues of $\tau(D_{\mathrm{Mt}})$ with two corresponding eigenvectors: α_{t1} and α_{t2}.

Jenkins and Matarić proposed spatial-temporal ISOMAP algorithm in 2004 [34]. Their method is comprehensive and defines the types of neighbors in detail. For different neighbors, the construction of the distance matrix is different in their method. In our method, we simply add temporal constraints on the construction of the distance matrix and strictly assume all the distances should be temporal related. This kind of method is simple but convenient for computation. Both Jenkins's method and our method are effective for describing the spatial-temporal characteristics of the sampled data points.

In current imitation learning, behaviors are special robotic movements in certain task-related situations. This means we can assume that the sampled data from demonstrations of one behavior always lie on the same manifold in the data space. The results of projecting data from the latent space to the data space must be on the same manifold as the demonstration. Therefore, it is reasonable to assume that there exists a relationship between the data in the data space and the latent space and it can be described as a function:

$$\theta_{vi}=f(x_i,W), \quad (6)$$

where x_i is a data point in the latent space and θ_{vi} is a corresponding data point in the original data space.

Therefore, $f(X,W)$ is designed as a generalized linear regression model:

$$\theta_{vi}=W\Phi(x_i). \quad (7)$$

$\Phi(x)$ is composed of R basis functions:

$$\Phi_i(x)=\exp\left(-\frac{(x_i-c_i)^2}{\Sigma_i}\right), \quad i=1,2,\dots,R, \quad (8)$$

where c_i is the center of the ith basis function and Σ_i is the bandwidth. The centers are uniformly distributed in the latent space and the bandwidth is designed for the basis functions to cover the latent space.

W is a $(D-1)*R$ matrix which projects the data from the latent space to the data space. However, Bishop has verified that the number of basis functions must typically grow exponentially with the dimensionality of the input space [35]. This means the advantage of dimension reduction in calculation and storage eventually arrives at a certain value as the number of the dimensionality increases. In Section 4, comparisons of results using different number of basis functions are given.

Assuming the projection matrix W is known, the probabilities of the distributions of the points in data space are

$$p(\theta_i\mid x_i,W,\beta)=\left(\frac{\beta}{2}\right)^D\exp\left(-\frac{\beta}{2}\|f(x,W)-\theta\|^2\right). \quad (9)$$

The log likelihood of the probability of the distribution of points in data space is the multiplication of the distribution probability of each point:

$$L(W,\beta)=\sum_{i=1}^{N}\ln p(t_i\mid x_i,W,\beta). \quad (10)$$

Maximizing the log likelihood function can be achieved by differentiating the log likelihood function with respect to W:

$$\sum_{i=1}^{N}(W\Phi(x_i)-\theta_i)\Phi(x_i)^T=0. \quad (11)$$

Rewrite (11):

$$W\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T=\sum_{i=1}^{N}\theta_i\Phi(x_i)^T. \quad (12)$$

Projection matrix W can be calculated from (11),

$$W=\sum_{i=1}^{N}\theta_i\Phi(x_i)^T*\left(\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T\right)^{\dagger}, \quad (13)$$

where $(\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T)^{\dagger}$ is the Moore-Penrosepseudo inverse matrix of $\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T$.

The sampled trajectory is projected from a 6-dimensional data space to a 2-dimensional space using the original ISOMAP algorithm and a modified ISOMAP algorithm in "feature extraction" block.

In the latent space, we have data points set X, which is a two-dimensional space. As stated in previous sections, in "dimension reduction," the temporal information is only used to calculate the neighborhood graph. But in the "behavior planning" stage, the temporal information should be combined into the model and set as the enquiry point.

Data points in the latent space follow (14):

$$x_i=f(t_i), \quad i=1,2,\dots,N. \quad (14)$$

Using Gaussian process [22], we can get a kernel method-based model of the demonstration in the latent space. The points on the two-dimensional plane are described as $x_i = \{x'_i, x''_i\}$, and one GP model is used in one dimension in the latent space. GP has been widely used [36–39] for representing the sampled data points because its robustness and nonparametric characteristics.

Assume the N two-dimensional data points in the two-dimensional latent space has the following probabilistic distribution:

$$\begin{aligned} p\left(\overrightarrow{z'} \mid \overrightarrow{x'}\right) &= \mathcal{N}\left(\overrightarrow{z'} \mid \overrightarrow{x'}, \beta^{-1} I\right), \\ p\left(\overrightarrow{z''} \mid \overrightarrow{x''}\right) &= \mathcal{N}\left(\overrightarrow{z''} \mid \overrightarrow{x''}, \beta^{-1} I\right). \end{aligned} \tag{15}$$

Take the calculation in the first dimension as an example: $p(\overrightarrow{z'}) = \mathcal{N}(\overrightarrow{z'} \mid 0, C_N)$, where the covariance matrix $C(n,m) = k(t_n, t_m) + \beta^{-1}\delta_{nm}$, and $\vec{z}$ is a vector of target values.

$k(t_n, t_m)$ is the kernel function. Normally,

$$k(t_n, t_m) = \theta_0 \exp\left\{-\frac{\theta_1}{2}\left\|t_n - t_m\right\|^2\right\} + \theta_2 + \theta_3 {t_n}^T t_m, \tag{16}$$

and x_n is considered as the timing step in the demonstration.

In the "Generation" stage, a new time step t_{enquiry} is given as an enquiry point and GP is used to calculate the corresponding data value z'_{enquiry}.

$$p\left(\left\{\overrightarrow{z'}, z'_{\text{enquiry}}\right\}\right) = \mathcal{N}\left(\left\{\overrightarrow{z'}, z'_{\text{enquiry}}\right\} \mid 0, C_{N+1}\right). \tag{17}$$

The covariance matrix is:

$$C_{N+1} = \begin{pmatrix} C_N & \vec{k} \\ \vec{k}^T & c \end{pmatrix}, \tag{18}$$

where $k = k(t_n, t_{\text{enquiry}})$ for $n = 1, 2, \ldots, N$.

Using Bayesian method, z'_{enquiry} is calculated using (19)

$$z'_{\text{enquiry}} = \vec{k}^T {C_N}^{-1} \overrightarrow{z'}. \tag{19}$$

Using the same method, z''_{enquiry} can be calculated using (20):

$$z''_{\text{enquiry}} = \vec{k}^T {C_N}^{-1} \overrightarrow{z''}. \tag{20}$$

In this part, the input is a data set including the sampled data points in a 6-dimensional joint space, and the output is a GP model for the trajectory in the latent space. Given an enquiry point (normally a timing point), the output of this model, which is (19) and (20), is a corresponding data point on the trajectory.

Using the following equation, we can project the data from the low-dimensional latent space to the original data space:

$$\theta_{vi} = W\Phi(x_i). \tag{21}$$

These data points will be used for robot to learn how to generate the required movement trajectories. In robotic imitation learning, robots need to fit the recorded movement trajectory with the model used in a generator [16, 40]. The fitting process is considered as learning a pattern in a generator.

In our system, we used the Dynamic Movement Primitives (DMP) [11], proposed by Ijspeert, as the pattern generator.

DMP is configured as

$$\begin{aligned} \tau\dot{z} &= \alpha_z\left(\beta_z(g - y) - z\right), \\ \tau\dot{y} &= z + f. \end{aligned} \tag{22}$$

g is the goal state, z is the internal state, f, an RFWR model, is calculated to record the dynamic of the demonstration and to guarantee convergence of the new generated trajectories, y is the position generated by the DMP differential equations, and $\dot{y}$ is the generated velocity correspondingly. α_z, β_z, and τ are the constants in this equation.

The fitting (or learning) is to train robots to learn the model:

$$f = \frac{\sum_{i=1}^{N} \Psi_i w_i v}{\sum_{i=1}^{N} \Psi_i}. \tag{23}$$

and v satisfies the following equation:

$$\begin{aligned} \tau\dot{v} &= \alpha_z\left(\beta_z(g - x) - v\right), \\ \dot{x} &= v. \end{aligned} \tag{24}$$

Ψ_i is a receptive basis function, which is distributed in the space:

$$\Psi_i = \exp\left(\frac{1}{-2{h_i}^2}(x - c_i)^2\right). \tag{25}$$

c_i is the center of the basis function, which is distributed in the space, and h_i is the bandwidth.

The target is to use the sampled points as x and use iterative learning method for robots to adapt the parameters w_i. After the learning, the parameters are fixed and do not need to change at the generation stage:

$$\begin{aligned} {\Delta_i}^{(n+1)} &= \exp\left(-\frac{1}{2}\left(x^{(n+1)} - c_i\right)^T D\left(x^{(n+1)} - c_i\right)\right), \\ {w_i}^{(n+1)} &= \lambda {w_i}^{(n)} + {\Delta_i}^{(n+1)}, \end{aligned} \tag{26}$$

The subscript $(n+1)$ denotes that this is the $(n+1)$ iteration, $x^{(n+1)}$ is the data point used to update the model at the $(n+1)$ iteration, and ${\Delta_i}^{(n+1)}$ is computed as the weighted distance between the data point $x^{(n+1)}$ and the center of the basis function, which is used to update the weight w_i.

3.2.2. Semantic Knowledge Learning. Recording and writing the characters are not enough for robots to interact with humans and robots should understand the semantic meaning of the motions and correlate the motion models and the semantic knowledge models automatically.

The general algorithm is shown in Figure 2.

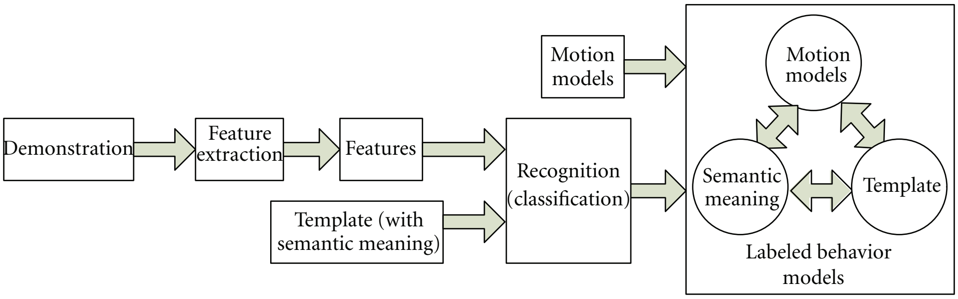


Figure 2: Learning of motion models and semantic knowledge.

The extracted feature of the demonstration is used to compare with the templates. The classification results automatically assign the semantic meaning of the template to the corresponding models. The learned motions should have relationship with its semantic-related templates. For example, the motions of writing characters should have relationships to the shapes or topologies of characters, and so on.

In this paper, ISAC is trained to write numbers and to learn the semantic meaning of the numbers automatically. As stated below, the demonstration is writing, the templates should be the shapes or topologies of the numbers. Correspondingly, in this part, the original ISOMAP method, which reflects the spatial topology of sampled data points, is used to train robots to learn semantic meaning of the motions. In order to reflect the overall spatial topology of the sampled data points, all of the neighbor points are considered as temporal neighbors. For simplicity, the modified ISOMAP algorithm can also be applied in this modeling part while the threshold value of temporal distance s is set as the size of the sampled data points.

Using the original ISOMAP algorithm, the demonstration of the motions of writing letters is projected onto a two-dimensional plane with corresponding projection matrices. The recorded trajectories in the latent space are normalized in the same scale. In this paper, the range of the x-axis is $[0, 1]$ and the range of the y-axis is also $[0, 1]$. The reason for normalization is obvious because the given demonstration by human teachers could have different scaling. In order to compare the demonstrations with the commands shown at the generation stage, these processed demonstrations should be normalized.

The technology of establishing models of the recorded trajectories writing numbers and recognizing the numbers based on the templates are not the concentrations of this paper; readers may be interested in other literatures to find many advanced word segmentation and recognition methods.

In this paper, an Optical Character Recognition (OCR) software tool, TesseractOCR (developed by Hewlett-Packard and currently maintained by Google), is used for ISAC to recognize comparing the knowledge model with the characters in the database. In practical application, the results of recognizing a single number are not good by using Tesseract-OCR. Therefore, the picture of a knowledge model is resized to normal size of a letter and placed behind a sentence "This is." Then the recognition is "This is **." By abandoning "This is," ISAC obtains the semantic knowledge of this recognized picture.

After recognition, the motion models are automated-assigned a semantic meaning and a corresponding template based on the recognition results.

The labeled motion models are stored in the "labeled behavior model" block.

In "Behavior Modeling" block, the projection matrices are calculated using a typical learning algorithm. The trajectories in the latent space and their corresponding projection matrices are stored in "Behavior Models" block. Given a command, ISAC parses the commands or recognizes the commands and converts them into actions in the "command analysis" block and retrieves the behavior models from the "Behavior Models" block. The "reconstruction" block projects the trajectory from the latent space to the joint space to generate new behaviors.

3.3. Generation. In the generation stage, the command is sent to robots and robots need to analyze the command and convert the command into actions with specified parameters. Then the required motion model is obtained by searching the semantic names of the actions with specified parameters. If the motion models are stored in the latent space, a reconstruction is needed to project the motion model from the latent space to the original data space; otherwise, the motion models will be used directly. At last, robots need to execute the motions to complete tasks.

3.3.1. Command Analysis. For most situations of imitation learning, robots are given a task-related situation. The starting states and the goal states are given in this situation and robots need to use learned behaviors to complete the task (achieving the goal state). In this paper, the speech command is used for robots to understand the task.

Using speech, robots needs to listen to the command from a human operator, recognize the required information, and convert the information into actions with specified parameters [41]:

$$\begin{aligned}\text{Action}(\text{parameter}) &+ \text{Action}(\text{parameter}) \\ &+ \cdots + \text{Action}(\text{parameter}).\end{aligned} \tag{27}$$

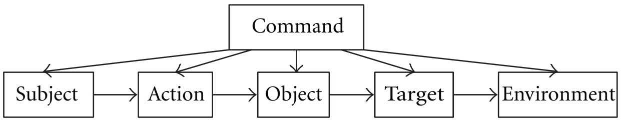


Figure 3: Command analysis.

Figure 3 displays the general method of analyzing the command.

Robots break down the commands into different parts by finding matching words in the lexicons using certain grammar rules. Subjects, actions, objects, targets, and environments are predefined in the lexicons. Using certain rules, the commands are converted into actions as shown in (7). This is a typical natural language processing method. Readers may refer to the book written by [42].

For the example of writing numbers, the lexicon design is

Subject: ISAC

Action: Write

Object: Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, and Ten.

The grammar design is Action+Object.

Receiving a command from a human teacher, ISAC extracts the "object" information from the commands and retrieves the corresponding behavior model from the "Labeled Behavior Models" block by searching the required "Object" in the behavior names. The implementation is using the Microsoft speech recognition library.

3.3.2. Reconstruction and Execution. If the motion models are stored in the latent space, using (6), the models can be projected from the latent space to the original data space.

In this paper, a GP-based model is used to describe the motion models in the latent space. Therefore, the (6) is rewritten as:

$$\theta = W\Phi(z), \tag{28}$$

where z is the data point obtained from the GP model given an enquiry point.

Then the required data in the joint space is obtained and robots move the actuator following the generated trajectory.

Using (28), required trajectories could be computed as θ_d.

Using forward kinematics, positions and orientations can be computed as

$$X_d = \text{Forward Kinematics } (\theta_d). \tag{29}$$

The target of generating a similar motion trajectory to complete a task is to minimize the error between the demonstrated trajectory and the generated trajectory.

We define the quadratic cost at each timing step is:

$$L_k = \left(X_{dk} - X_{gk}\right)^T W_k \left(X_{dk} - X_{gk}\right) \tag{30}$$

X_d is a desired trajectory (a demonstrated trajectory in this paper), X_g is a generated trajectory, and k is the timing step.

L represents the weighted error between the demonstrated trajectory and the generated trajectory at timing step k. The target is to minimize the overall cost:

$$\Phi(N) + \sum_{k=1}^{N-1} L_k. \tag{31}$$

while $\Phi(N)$ is the terminal cost, which is normally defined as:

$$\Phi(N) = \left(X_{dN} - X_{gN}\right)^T W_N \left(X_{dN} - X_{gN}\right). \tag{32}$$

For simplicity, in our algorithm, W_k and W_N are defined as a unity diagonal matrix.

The control process is an integration of sensing and planning. In this paper, we do not focus on the low level actuator control. Because the regulators on ISAC are commercial devices just like "black-boxes", we assume that a regulator can automatically adjust control output to achieve the control goals when a required reference input is given.

The initial position and orientation are computed by:

$$X_0 = \text{Forward Kinematics } (\theta_0). \tag{33}$$

At timing step k, θ_{sk-1} is the sensed joint angles at timing step $k-1$ and assume the θ_{gk} is planned based on the current sensing information:

$$X_{gk} - X_{gk-1} = J * \left(\theta_{gk} - \theta_{sk-1}\right), \tag{34}$$

where X_{gk-1} is computed by:

$$X_{gk-1} = \text{Forward Kinematics } (\theta_{sk-1}), \tag{35}$$

and J is the Jacobian matrix.

The target is to minimize

$$L_k = \left(X_{dk} - X_{gk}\right)^T W_k \left(X_{dk} - X_{gk}\right). \tag{36}$$

Rewrite (36):

$$\begin{aligned} L_k = &\left(X_{dk} - J\left(\theta_{gk} - \theta_{sk-1}\right) + X_{gk-1}\right)^T \\ &\times \left(X_{dk} - J\left(\theta_{gk} - \theta_{sk-1}\right) + X_{gk-1}\right). \end{aligned} \tag{37}$$

Minimize this cost function by differentiating L_k with respect to θ_{gk} and set the derivative to zero, we get

$$\begin{aligned} \theta_{gk} = &\frac{1}{2}\left(X_{dk} - X_{gk-1} + J\theta_{sk-1}\right)^T \\ &- \frac{1}{2}J^T\left(X_{dk} - X_{gk-1} + J\theta_{sk-1}\right). \end{aligned} \tag{38}$$

At each timing step, θ_{gk} is given to the regulator as reference input for low-level actuator control.

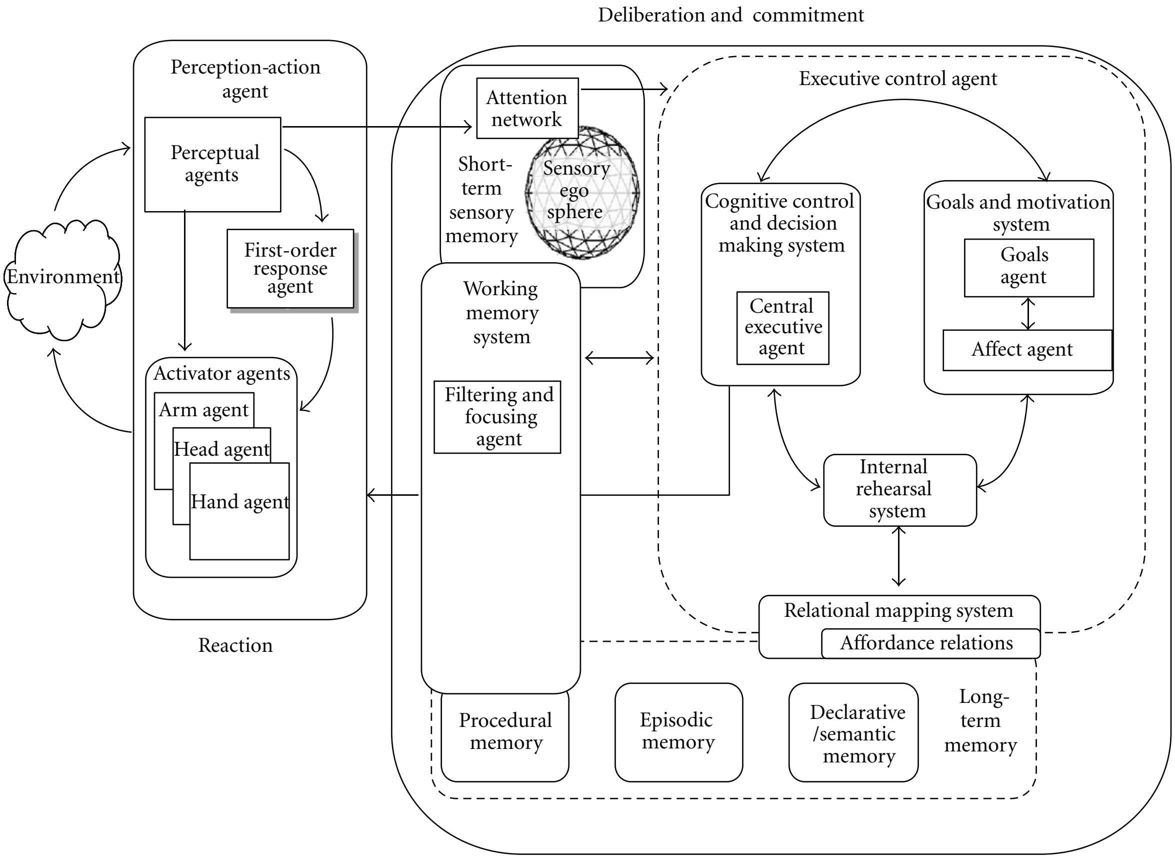


Figure 4: ISAC cognitive architecture.

3.4. *Implementation.* This framework is implemented on a cognitive architecture, named ISAC cognitive architecture, developed by the Center for Intelligent Systems of Vanderbilt University [43–45].

Figure 4 displays the system design of the ISAC cognitive Architecture.

3.4.1. Perceptual Agents (PA). The PA obtains the sensory information from environment. Normally, encoders on the joints of the robot, cameras on the head of the robot, and the force feedback sensor on the wrist of the robot are implemented in this agent.

3.4.2. Short-Term Memory (STM). The obtained information is sent to and stored in the STM. The Sensory Ego Sphere (SES) is implemented in the STM, which performs spatio-termporal coincidence detection, mediates the salience of each percept, and facilitates perceptual binding.

3.4.3. Working Memory System (WMS). The WMS stores the task-related information in chunks. This component is especially important in the generation stage.

3.4.4. Central Executive Agent (CEA). The CEA provides central processing, decision making, and control policy generating for different task goals which is stored in the Goals Agent (GA). In hierarchy architecture, this component accesses all of the sensed information and makes decision for tasks.

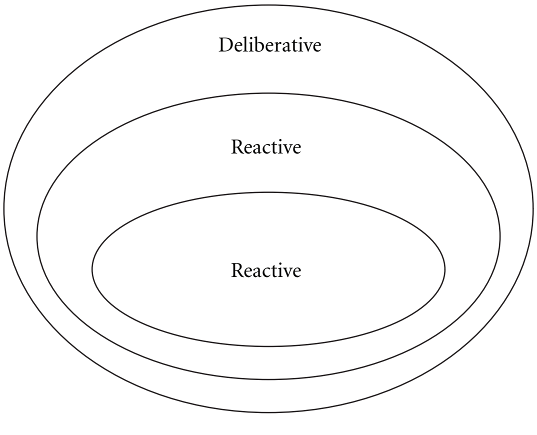


Figure 5: Work loops.

3.4.5. Goal Agent (GA). Correspondingly, the GA stores the motivations or goals of tasks in situations.

3.4.6. Long-Term Memory (LTM). The LTM stores the memory especially the knowledge for long term use. Procedural,

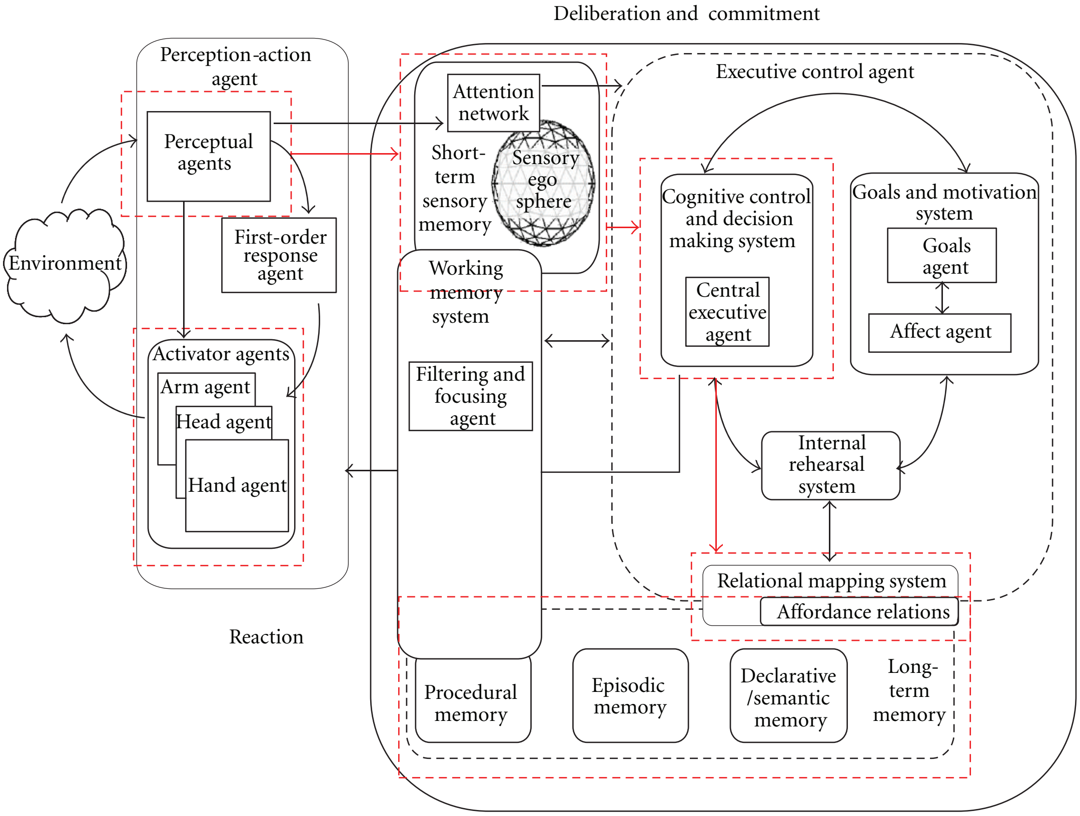


Figure 6: Learning stage in ISAC cognitive architecture.

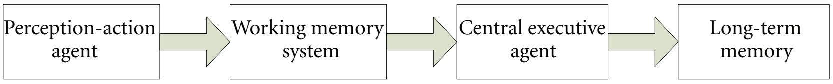


Figure 7: Work loop of the learning stage.

semantic, and episodic knowledge are stored in this component. In imitation learning, the learned skill or knowledge is stored as procedural and episodic knowledge using a mathematical model.

3.4.7. Internal Rehearsal System (IRS). The IRS evaluates the results of the decisions made from the CEA through internal rehearsal.

3.4.8. Work Loops. Using this architecture, we can develop three work loops: reactive, routine, and deliberative.

Reactive Loop is inside the perception-action agent. The perceptual agent collects the sensory information from the environment. Using the first-order agent, necessary actions are taken by the actuator agent to affect the environment and the robotic body. This control loop is used for robots to process the emergency or unexpected change in the environment.

The Routine Loop is within the the Perception-Action Agent, filtering and focusing Agent, the STM, and the WMS. This loop completes routine tasks which are well defined in the WMS. The robot obtains the task-related information from the WMS and sends this information to the actuators through Filtering and Focusing Agent. Actuators are driven by the received information to complete tasks. The Routine Loop also involves the Reactive Loop to avoid the unexpected changes in the environment.

The Deliberative Loop is used for robots to learn new behaviors or skills through modeling, knowledge coupling, and so forth, and to complete new tasks or select behaviors to complete tasks using reasoning, decision making, and so forth. The CEA is the central component in this loop. It retrieves the stored knowledge from the LTM, receives the environmental information from the STM and the WMS, and uses the IRS to evaluate current situation to make decisions or establish models for the sensed information. When the decision is made, the task-related information is sent to the WMS and the system will use the Routine Loop to complete the task. The Deliberative Loop involves the Reactive Loop and the Routine Loop. Our system is largely based on the Deliberative Loop.

Figure 5 displays the relationships among the tree work loops.

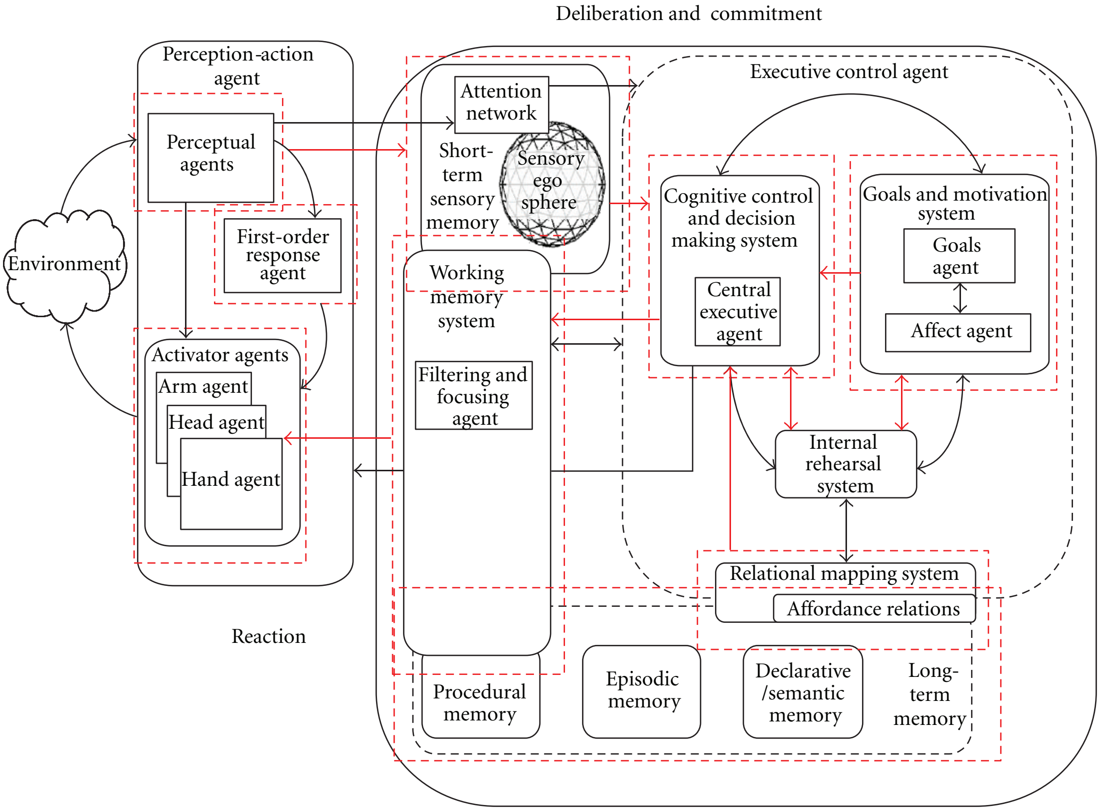


FIGURE 8: Generation stage in ISAC cognitive architecture.

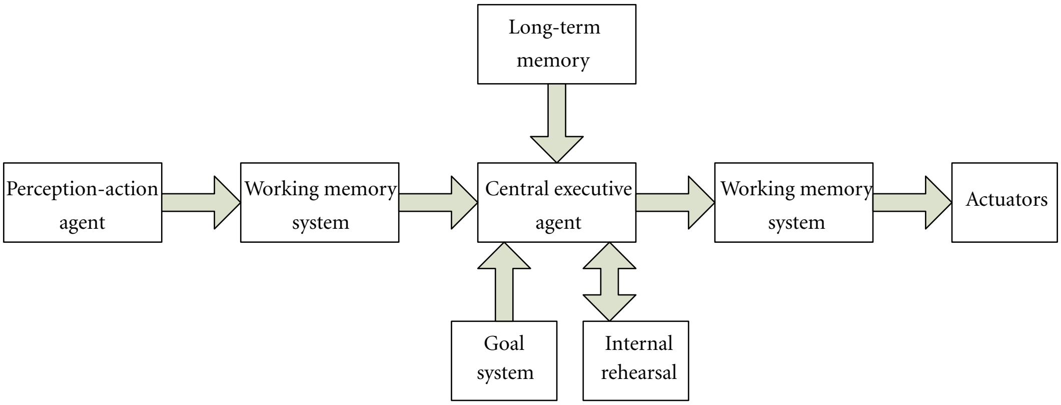


FIGURE 9: Work loop of the generation stage.

3.4.9. Learning Stage. At the learning stage as shown in Figure 6, ISAC collects the information from the encoders using the PA and sends the sensory information to the CEA. The CEA obtains the original ISOMAP algorithm and the modified ISOMAP algorithm from the LTM and calculates the motion models and the knowledge models. Using the RMS, ISAC establishes the labeled behavior models upon the prelearned semantic knowledge and stores the models in the LTM.

The work loop could be displayed as in Figure 7.

3.4.10. Generation Stage. At the Generation Stage as shown in Figure 8, given a speech command, ISAC collects the speech information using the PA, and sends the speech information to the CEA through the STM. By analyzing the speech command, the CEA generates the corresponding actions with specified parameters. The required behavior model is obtained from the LTM through searching the RMS. Upon the obtained behavior model, the CEA plans the motions according to the goal and sends the motion information to the WMS. The WMS stores the task-related

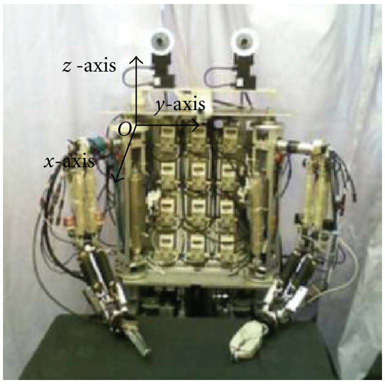


Figure 10: ISAC robot.

Figure 11: Letters for demonstrations.

information and sends the control commands to the AA to execute the motions.

The work loop could be displayed as in Figure 9.

4. Experimental Results

A humanoid robot, named ISAC, is used to validate out proposed system (as shown in Figure 10). ISAC is a stationary pneumatic driven humanoid robot, which has seven Degrees-of-Freedom (DOFs) on each of its arm (including a freedom (open and close) for the end effector). In this system, we only used the right of ISAC to demonstrate the movement trajectories of writing and to write the required numbers. The pen is always grasped using the end-effector, and we only use six DOFs of the right arm of ISAC. Two cameras mounted on the robot are used for ISAC to observe the environment, and we developed an OpenCV-based program to capture and process the images obtained from the cameras. A personal computer, with a 1 GHz CPU, is used to control the arm of ISAC, a personal computer, with a 2.4 GHz CPU, is used to process the images, and a laptop, with a 2.4 GHz CPU, is used to store the semantic knowledge models and the movement trajectory models.

ISAC is shown how to write letters by manually moving its right arm as shown in Figure 12. Figure 11 displays the letters used in the demonstrations. The topologies of the movements of writing letters in the Cartesian space are also the same as shapes of the letters.

The collected data is projected onto a 2-dimensional plane using the original ISOMAP algorithm and the modified ISOMAP algorithm. Figure 13 displays the obtained model using the original ISOMAP algorithm, and Figure 14 displays the obtained model using the modified ISOMAP algorithm.

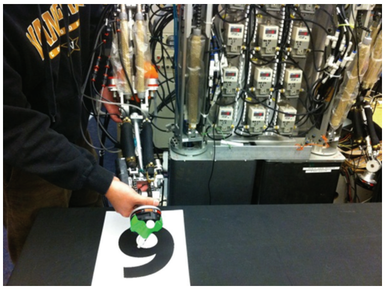


Figure 12: Demonstrations.

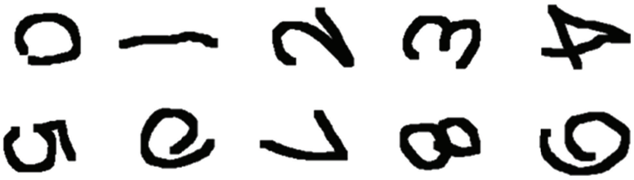


Figure 13: Dimension reduction results of the original ISOMAP algorithm.

In practical application, in order to use the dimension reduction results in the recognition part, the image on the 2-dimensional plane has been dilated.

In Figure 13 the dimension reduction results display the shapes and topologies of the distributions of the sampled joint angles from the demonstrations in the latent space. From this figure, the shapes and the topologies of the data distributions are similar to the real letters on the paper and the movement of the end-effector for writing letters in the Cartesian space. It is necessary to emphasize here that without using the kinematics model which calculates the position of the end-effector in the Cartesian space using the joint angles, the dimension reduction results can still get approximate descriptions of the letters on the paper.

In Figure 14, the motion models are represented in the latent space. These models are obtained using the modified ISOMAP algorithm. From the experimental results, we can find two features of the models using this algorithm. (1) Each trajectory does not overlap with itself; (2) each trajectory, which is generated automatically using the modified ISOMAP, always starts from one side and ends at another side. The second feature is guaranteed by the definition of the neighbors in the algorithm. Intuitively, because neighbors are defined as temporal neighbors, the temporal distance of the first point and the last point is the largest in the distance matrix. Therefore, the algorithm always put the starting point and the ending point at two opposite sides of the graph.

Through recognition, the knowledge models in Figure 13 are recognized with the pre-learned numbers using the Tesseract-OCR.

In practical application, the pictures in Figure 13 should be preprocessed in order to be compatible with the Tesseract-OCR. There are several steps of preprocessing.

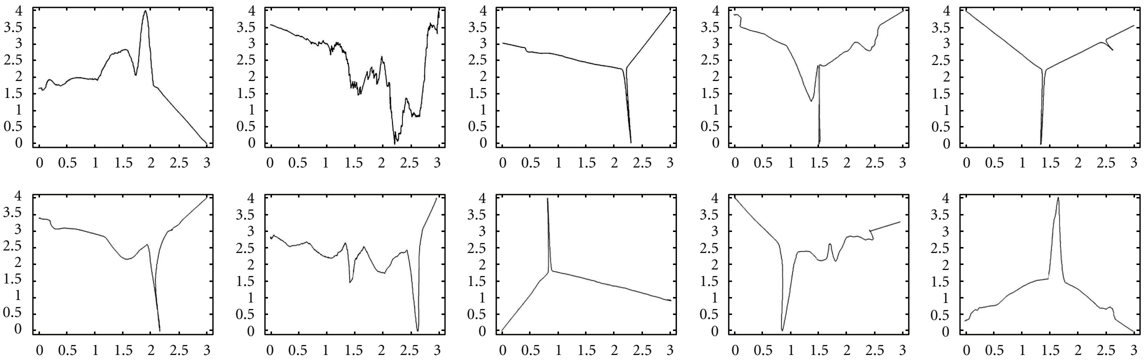


Figure 14: Motion models in the latent space using the modified ISOMAP algorithm.

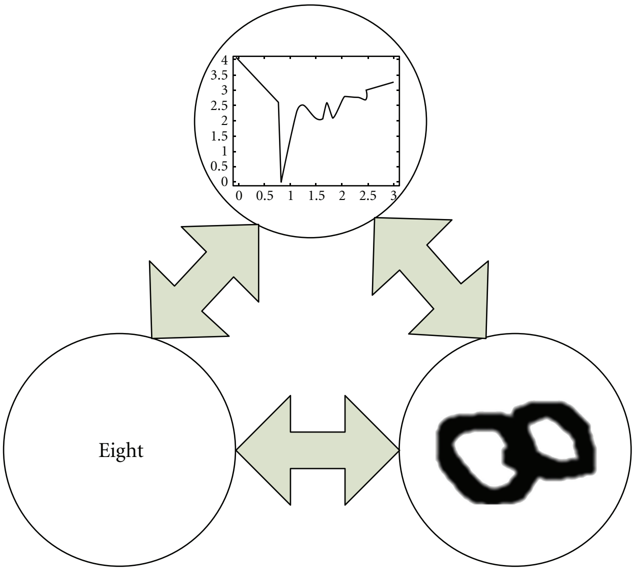


Figure 15: Labeled Behavior Models.

(1) Picture i is rotated 0 degree, the result is picture i_1.

(2) Picture i is rotated 90 degree, the result is picture i_2.

(3) Picture i is rotated 180 degree, the result is picture i_3.

(4) Picture i is rotated 270 degree, the result is picture i_4.

(5) Picture i is flipped horizontally, the result is picture i_5.

(6) Picture i_5 is rotated 90 degree, the result is picture i_6.

(7) Picture i_5 is rotated 180 degree, the result is picture i_7.

(8) Picture i_5 is rotated 270 degree, the result is picture i_8.

The obtained eight pictures are all recognized using the Tesseract-OCR. If the recognized result is included in the predefined white list: $\{0, 1, \ldots, 9\}$, it is accepted.

If the recognized result is (6) or (9), it needs to be further processed. Our method is to determine the starting point of writing (6) or (9). If the starting point is near the edge of the image, it is (6); otherwise, it is (9).

Upon the recognition results, the labeled behavior models are established. Figure 15 is an example of the models.

Upon receiving a command from humans, ISAC analyzes the command and converts the command into the actions with specified parameters: Write (Six) as shown in Figure 16. The required motion model is obtained from the labeled behavior models and ISAC executes the motions.

Figure 17 displays the numbers written by ISAC on the papers.

5. Discussion

In this paper, we proposed a framework for robots to learn the motion models and semantic knowledge models simultaneously and only one dataset from the demonstrations is used for both learning stages.

In the current imitation learning frameworks, the motion learning has been highlighted for robots to learn to complete some interesting tasks.

Some researchers are working on incorporating concepts and ideas from cognitive science into robotics research. A typical application using cognitive architectures to implement cognitive processes or cognitive control loops for robotic control and learning [46].

If we consider the whole robotic learning frameworks as hierarchy architectures, the emergent problem is that there seems a gap exists between the utilization of cognitive architecture and the motion learning. As we know, the reasoning and the planning in the cognitive architecture are often implemented in a symbolic way and traditional Artificial Intelligence (AI) methods are used. Therefore, how to connect the symbolic representation and the mathematical models-based motion models?

In our paper, a framework is proposed, in which the connection is based on the natural language processing, which labels the motion models with suitable behavior names and analyzes the command in the cognitive architecture with the behavior names. In this paper, we further propose to train robots to learn the semantic (or symbolic) knowledge by using the same dataset from the demonstrations automatically, which enhance our proposed framework.

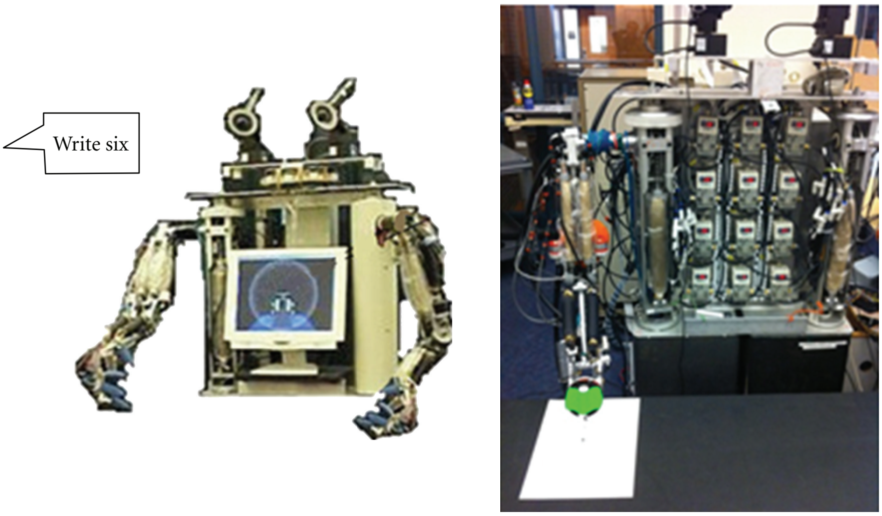


Figure 16: Command analysis and generation.

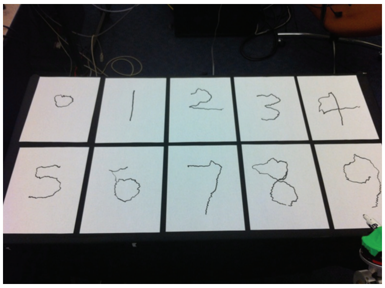


Figure 17: Letters written by ISAC.

As we know, when humans see, listen, and feel the behaviors from other humans, we can relate what we see, listen, and feel to our learned procedural, episodic, and semantic knowledge. The framework proposed in this paper is inspired from the daily cognitive brain work of humans.

6. Future Study and Conclusion

For the application of "writing," using the same methods in this paper (dimension reduction using the original and modified ISOMAP algorithm and a letter recognition technology) robots can learn how to write letters and relate the motion models and knowledge models to their corresponding semantic knowledge models.

In other areas, for example, "music playing," robots can learn how to play music (hitting drums, playing guitar, and playing piano) and relate the motions required to play music to their corresponding semantic knowledge models. The ISOMAP algorithms may not complete such learning in these areas. However, readers can simply find that the tempos of hitting drums are in correspondence to the tempos of moving hands up and down. If the tempos of the sound of hitting drums can be extracted as the templates and the tempos of moving hands can be extracted as the knowledge model, this framework can also be used for robots to learn the motions of play music and the semantic meaning of these motions simultaneously and automatically.

The crucial point of the application of this framework is to find the features of the motions which are in correspondence with the inner features of behavior which is strongly related to the semantic models.

This paper proposes a framework for robots to learn the motion models and semantic knowledge models simultaneously using one data set from the demonstrations. A modified ISOMAP algorithm is used for robots to extract semantic information from the demonstrations. The implementation is on a cognitive architecture with several extensions of current algorithms. Semantic analysis of the command is also implemented in this framework. The experiments are carried out on a humanoid, and the experimental results demonstrate the effectiveness of this framework.

References

[1] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE journal of robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.

[2] R. Brooks, "How to build complete creatures rather than isolated cognitive simulators," in *Architectures for Intelligence*, K. VanLehn, Ed., pp. 225–239, Lawrence Erlbaum Associates, New York, NY, USA, 1991.

[3] A. Sloman and J. Chappell, "The altricial-precocial spectrum for robots," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCA '05)*, pp. 1187–1192, Edinburgh, UK, 2005.

[4] A. Stoytchev, "Toward learning the binding affordances of objects: a behavior-grounded approach," in *Proceedings of the AAAI Symposium on Developmental Robotics*, pp. 21–23, 2005.

[5] S. Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems*, M. J. M. C. Mozer and T. Petsche, Eds., pp. 1040–1046, The MIT Press, Cambridge, Mass, USA, 1997.

[6] M. Uchiyama, "Formation of high speed motion pattern of mechanical arm by trial," *Transactions, Society of Instrument and Control Engineers*, vol. 19, pp. 706–712, 1978.

[7] C. Atkeson and J. McIntyre, "Robot trajectory learning through practice," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1737–1742, San Francisco, Calif, USA, 1986.

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[9] A. Billard, "Learning motor skills by imitation: a biologically inspired robotic model," *Cybernetics and Systems*, vol. 32, no. 1-2, pp. 155–193, 2001.

[10] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[11] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15, pp. 1547–1554, The MIT Press, 2003.

[12] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zollner, and M. Bordegoni, "Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm," in *Proceedings of the 9th International Symposium of Robotics Research (ISRR '99)*, pp. 229–238, Snowbird, Utah, USA, October 1999.

[13] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

[14] T. Inamura, M. Inaba, and H. Inoue, "Acquisition of probabilistic behavior decision model based on the interactive teaching method," in *Proceedings of the 9th International Conference on Advanced Robotics,*, pp. 523–528, 1999.

[15] R. M. Voyles and P. K. Khosla, "A Multi-agent system for programming robots by human demonstration," *Integrated Computer-Aided Engineering*, vol. 8, no. 1, pp. 59–67, 2001.

[16] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation*, pp. 1398–1403, Washington, DC, USA, May 2002.

[17] I. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 1986.

[18] D. J. Bartholomew, "The foundations of factor analysis," *Biometrika*, vol. 71, no. 2, pp. 221–232, 1984.

[19] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[20] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[21] C. K. I. Williams, "On a connection between kernel PCA and metric multidimensional scaling," *Machine Learning*, vol. 46, no. 1–3, pp. 11–19, 2002.

[22] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[23] C. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*, pp. 63–71, The MIT Press, Cambridge, Mass, USA, 2004.

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

[25] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: an O (n) algorithm for incremental real time learning in high dimensional space," in *Proceedings of The 17th International Conference on Machine Learning*, pp. 288–293, Stanford, Calif, USA, 2000.

[26] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using Gaussian mixture models," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, p. 233, 2007.

[27] H. Abdi, "A neural network primer," *Journal of Biological Systems*, vol. 2, no. 3, pp. 247–283, 1994.

[28] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, The MIT press, 1998.

[29] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robotis*, pp. 1–20, Karlsruhe, Germany, 2003.

[30] P. Dyer and S. R. McReynolds, *The Computation and Theory of Optimal Control*, Academic Press, 1970.

[31] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: a path integral approach," in *IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 2397–2403, 2010.

[32] E. A. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[33] H. Tan and K. Kawamura, "A computational framework for integrating robotic exploration and human demonstration in imitation learning," in *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 2501–2506, Anchorage, Alaska, USA, 2011.

[34] O. C. Jenkins and M. J. Matarić, "A spatio-temporal extension to isomap nonlinear dimension reduction," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, p. 56, July 2004.

[35] C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM: the generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.

[36] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 367–372, September 2008.

[37] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proceedings of the Robotics: Science and Systems (RSS '06)*, The MIT Press, 2006.

[38] A. Shon, K. Grochow, A. Hertzmann, and R. Rao, "Gaussian process Cca for image synthesis and robotic imitation," Tech. Rep. UW-CSE-TR-2005-06-02, University of Washington CSE Department, 2005.

[39] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 255–260, October 2010.

[40] H. Tan, E. Erdemir, K. Kawamura, and Q. Du, "A potential field method-based extension of the dynamic movement primitive algorithm for imitation learning with obstacle avoidance," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 525–530, Beijing, China, 2011.

[41] H. Tan, Q. Du, and N. Wu, "A Framework for cognitive robots to learn behaviors through imitation and interaction

with humans," in *Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pp. 235–238, New Orleans, La, USA, 2012.

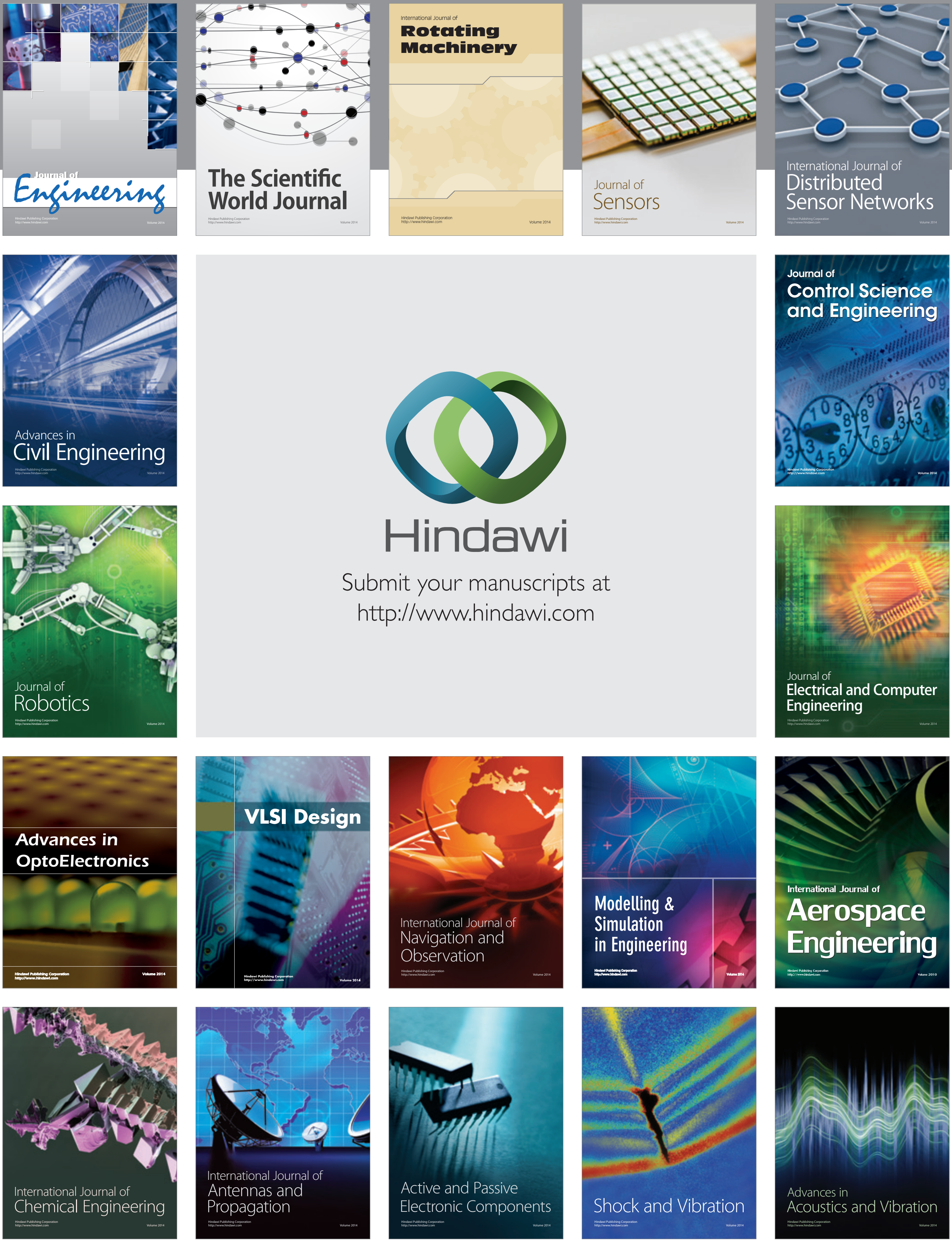
[42] S. J. Russell and P. Norvig, *Artificial Intelligence : A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2010.

[43] K. Kawamura, R. Peters II, R. Bodenheimer et al., "Multiagent-based cognitive robot architecture and its realization," *International Journal of Humanoid Robotics*, vol. 1, pp. 65–93, 2004.

[44] K. Kawamura, S. M. Gordon, P. Ratanaswasd, E. Erdemir, and J. F. Hall, "Implementation of cognitive control for a humanoid robot," *International Journal of Humanoid Robotics*, vol. 5, no. 4, pp. 547–586, 2008.

[45] H. Tan, "Implementation of a framework for imitation learning on a humanoid robot using a cognitive architecture," in *The Future of Humanoid Robots: Research and Applications*, R. Zaier, Ed., pp. 189–210, InTech Open Access Publishing, 2012.

[46] H. Tan and C. Liang, "A conceptual cognitive architecture for robots to learn behaviors from demonstrations in robotic aid area," in *Proceedings of 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1248–1262, Boston, Mass, USA, 2011.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration