# Learning Interaction Protocols using Augmented Baysian Networks Applied to Guided Navigation

Yasser Mohammad and Toyoaki Nishida

*Abstract*— **Research in robot navigation usually concentrates on implementing navigation algorithms that allow the robot to navigate without human aid. In many real world situations, it is desirable that the robot is able to understand natural gestures from its user or partner and use this understanding to guide its navigation. Some algorithms already exist for learning natural gestures and/or their associated actions but most of these systems does not allow the robot to automatically generate the associated controller that allows it to actually navigate in the real environment. Furthermore, a technique is needed to combine the gestures/actions learned from interacting with multiple users or partners. This paper resolves these two issues and provides a complete system that allows the robot to learn interaction protocols and act upon them using only unsupervised learning techniques and enables it to combine the protocols learned from multiple users/partners. The proposed approach is general and can be applied to other interactive tasks as well. This paper also provides a real world experiment involving 18 subjects and 72 sessions that supports the ability of the proposed system to learn the needed gestures and to improve its knowledge of different gestures and their associations to actions over time.**

## I. INTRODUCTION

Personal robots are expected to live within human society in near future. These robots should be operated with untrained users who may even have some limited interaction capabilities like the elderly and autistic children. Being able to understand and act upon natural gestures is a major advantage for such robots [1]. This ability can be utilized in a programming by demonstration context to endue the robot with more and more behavioral capabilities [2].

One example of a situation in which the ability to understand natural gesture is important is the guided navigation task. In this task the robot navigates in some environment that it cannot sense accurately using information from a human partner that is transferred via natural modalities like gestures. In a previous study we have shown that in such an environment, gestures are at least as effective as verbal communication [3]. In some cases (e.g. when the workspace is noisy) the verbal channel is not even available.

The most complex situation concerning this scenario is when the robot needs not only to learn what to do when receiving a gesture but to also learn the number and types of gestures that are used by its partner as well as the controllers that achieve the required movements based on these gestures. Solving these three problems simultaneously (learning action and gesture spaces and their associations) is a difficult problem and most available literature tries to simplify the problem.

For example Calderon and Hu [4] introduced a system for learning the reproduction of the path followed by the hand of a human by observing the motion. The action segmentation problem was ignored in this work as the whole movement is considered as a single action. Most of the research in the area of learning by demonstration have the same limitation (e.g. [5] [6]) as noted in [7] and [8]. Learning actions from a continuous stream of motion data was studied in the recent years. Ogata et. al. [9] developed a long term, incremental learning system using neural networks but for a single task. Takano and Nakamura [10] developed a system for automated segmentation, recognition and generation of human motions based on Hidden Markov Models. The number of primitives (commands/actions) have to be known priori. Kadone and Nakamura [11] developed a system for automated segmentation, memorization, recognition and abstraction of human motions based on associative neural networks. The main limitation of this system is that the abstracted motion representation can only be used for subsequent motion recognition, and cannot be used for motion generation. Kulic et. al. [12] presented a system that can incrementally learn body motion and generates a hierarchy of HMMs that can be used subsequently for generation and recognition. The main limitation of this system for our approach is that it assumes that the actions are already segmented into observations.

Mohammad et. al. [13] proposed a system that can simultaneously learn gestures, actions and their associations by casting the problem as a constrained motif discovery problem [14] and then providing novel algorithms to solve it. The final output of the system was a probabilistic network of the interaction protocol capable of predicting the behavior of a human actor with 95.2% accuracy. The main limitation of this work was that the learned system is unable to actually actuate the robot as there was no mechanism to *learn* the controllers required to do the learned actions. Another problem was that there is no proposed method to accumulate the gestures and actions learned from interacting with multiple partners. A third problem was that the proposed system was validated using small number of interactions.

This work resolves these limitation by providing a mechanism to construct the actual controller of the robot and a simple algorithm for combining the probabilistic networks learned from multiple partners. The paper also reports an

This work was partially supported by Kyoto University GCOE fund for young researchers

Y. Mohammad is with Faculty of Engineering, Electrical Engineering Department, Assiut University, Assiut, Egypt yasserm@aun.edu.eg

T. Nishida is with the Department Intelligence Science and Technology, Kyoto University, Kyoto, Japan nishida@i.kyoto-u.ac.jp

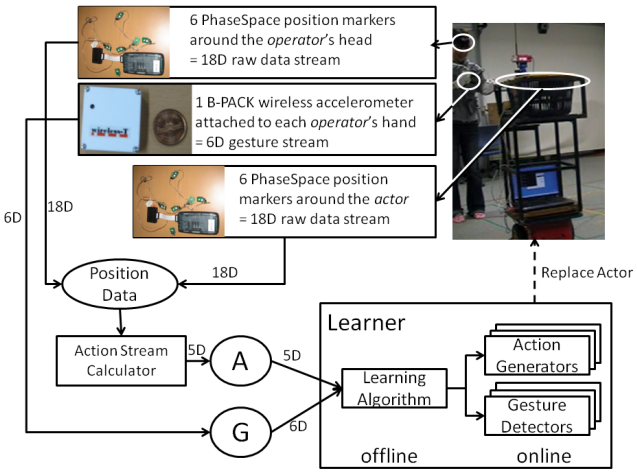



Fig. 1. Overview of the guided navigation scenario showing the role of the three agents, the sensors used and the overall view of processing steps.

evaluation experiment with 18 subjects involving 72 sessions that confirms the ability of the system to learn guided navigation from a single trial as well as its ability to accumulate learned gestures and actions from multiple human partners.

The rest of this paper is organized as follows: The following section introduces the guided navigation scenario and describes the sensors used in the experiments. Section III provides an overview of the complete system and briefly describes the D&A algorithm proposed in [13] to learn an offline version of the interaction protocol. Section IV details the controller generation algorithm while section V details the algorithm used to combine learned interaction protocols. Section VI presents the evaluation experiment, section VII discusses its results and section VIII describes some of the limitations of the proposed approach. The paper is then concluded.

## II. Guided Navigation Scenario

The guided navigation scenario used in this work involves three agents. The *operator* agent is a human that uses free hand gestures to instruct the *actor* agent (a robot in our case) to follow some path, avoid some obstacle, etc. The third agent is called the *learner*. The learner starts by *watching* some actor–operator interactions using a motion capture system and then uses unsupervised learning techniques to model both operator's gestures and actor's actions. It then builds a controller for each learned action and a detection mechanism for each learned gesture. The *learner*'s goal is to be able to act in *actor*'s position.

The inputs to the learning robot are two multidimensional time series:

1) Gesture stream ($G$) representing the motion of the guiding human. In this work we use wireless accelerometers called B-PACK [15]. Two accelerometers attached to the middle finger tips of the operator's two hands which results in a six dimensional gesture stream.
2) Action Stream ($A$) representing the motion of the guided human/robot. In this work we use a motion capture system called PhaseSpace[16] to get this action stream.

To calculate the action stream we first capture the position of the actor and the operator using the PhaseSpace motion capture system which can determine the 3D position of a set of markers. Six markers where attached to the head of the operator and six markers around the actor. This constitutes a 36 dimensional raw data stream. The 2D absolute positions of the operator and actor in the floor was calculated from this raw data by first estimating the center of each agent using each available marker attached to it and averaging to reduce the effect of noise and missing marker positions. This was necessary because of the high rate of missing data from the motion capture system.

The action stream $A$ was calculated from these positions by calculating the following:

- 2D distance and angle ($r_o$,$\theta_o$) between the actor and the operator in the operator's coordinate system.
- 2D position of the actor in the plan of motion ($x_a$,$y_a$) in Cartesian coordinates.
- Orientation of the actor in the plan relative to the direction of its starting point ($\theta_a$).

These features constitute the 5-dimensional action stream ($A$). These features were selected to cover both operator-centric and robot-centric direction and orientation. Fig. 1 shows the sensors used and the calculation of action and gesture streams. The goal of the *learner* in the guided navigation scenario is to use only $A$ and $G$ with no prior knowledge to learn how to act in the *actor*'s role and then to actually replace the actor in future sessions.

If the numbers of gestures and actions as well as the controllers to achieve these actions were already known, the guided navigation problem becomes a simple association problem. The main challenge that the proposed approach is trying to face is solving this problem with no prior knowledge of action and gesture numbers, durations or meanings, no prior knowledge of the characteristics of the action and gesture streams, and no known motor primitives that can achieve the required actions. It should be noted that both action and gesture streams are in the sensor space of the *learner*. For this reason the *learner* needs also to learn the mapping between the actions it perceives (in the sensor space) and commands to its motors (speeds of two DC motors in differential drive arrangement in our experiments) to actually realize these actions.

The solution proposed in this paper requires only unsupervised learning techniques, requires no input other than the unmarked continuous action and gesture streams, and does not rely on any characteristics of these streams that are specific to the guided navigation case.

The main motivation behind this generality is to have the proposed solution applicable to other tasks (e.g. picking objects, arranging a table, etc) and communication modalities (e.g. short verbal commands, spontaneous body movements, etc).

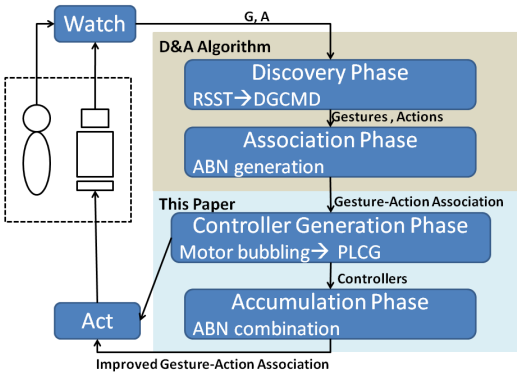



Fig. 2. Overview of the learning system. The D&A algorithm is a slightly modified version of [13]

The main reason for using the relatively simple guided navigation problem in this exploratory study is to focus on the learning algorithm rather than the complex details of the task to be learned and to confirm the applicability of the proposed algorithm in a simple case before applying it to more complex situations involving for example feedback from the actor or more teammate oriented interactions. In the future, the proposed approach will be applied and tested in more complex interactions to confirm the generality of the approach.

## III. The Learning System

The learning robot watches a set of interactions collecting $G$ and $A$ for each of them. It then analyzes the collected corpus to build its own controllers for guided navigation as will be explained in this section. The problem is decomposed into four different problems. Firstly, the robot needs to *discover* relevant action and gesture patterns in $A$ and $G$ without prior knowledge of their types, lengths or occurrence probabilities. This is called the discovery phase. Secondly, the robot needs to *associate* discovered gestures with the needed actions. This is called the association phase. Thirdly, the robot needs to generate actual controllers that can achieve the learned actions and associate them with the discovered actions. This is called controller generation phase. Finally, the robot combines the gestures and actions it learned from multiple interactions in the accumulation phase. Mohammad et al. [13] provided an algorithm for the first two phases. This system will be called the $D\&A$ algorithm in this work. In this work we utilize $D\&A$ and will explain it briefly in this section. The following sections will detail controller generation and accumulation phases.

Algorithm 1 depicts the $D\&A$ algorithm as utilized in this paper. The first two steps are applied to every dimension in $A$ and $G$ separately ($n_G$ and $n_A$ dimensions respectively).

Firstly, the points in the time series in which there is high probability that the underlying process is changing dynamics are discovered using the Robust Singular Spectrum Transform (RSST) [14]. The $n_G + n_A$ time-series representing the change scores are then combined using g-Causality

**Algorithm 1** Simplified version of $D\&A$ Algorithm

1: Find change scores for gestures and actions using RSST.
2: Combine change scores to form occurrence constraints $C^g$ and $C^a$.
3: Use DGCMD algorithm to discover Gesture Models $G_i^{hmm}$, Action Models $A_j^{hmm}$, Gesture Occurrences $O^g(t)$, and Action Occurrences $O^a(t)$.
4: Use correlations between $O^g(t)$ and $O^a(t)$ and g-Causality maximization to induce the final $ABN$

maximization. The main idea behind this procedure is to add to the change score of every dimension the nearby change scores corresponding to dimensions that are estimated to be related causally to this dimension. These two steps generate a new $n_G$ dimensions time series called $C_i^G$ and a new $n_A$ dimensions time series called $C_i^A$ representing the change score at every point for the gesture and actions streams.

The third step is to discover recurrent patterns around these change points (called motifs) using the Distance Graph Constrained Motif Discovery (DGCMD) algorithm presented in [13] which utilizes $C_i^G$ and $C_i^A$ to reduce the search space for recurrent patterns. The output of this step is a set of gesture and action primitives represented in the gesture and action dimensions using Hidden Markov Models as well as the mean of every one of these primitives.

The final step in the D&A system is the association step in which the system generates an Augmented Baysian Network (ABN) representing the relation between gestures and actions (interaction protocol). The ABN is a normal Baysian Network with added two values to each link (connecting nodes $n_1$ and $n_2$) called $\mu_{n_1-n_2}$ (mean of the delay between $n_1$ activation and $n_2$ activation) and $\sigma^2_{n_1-n_2}$ (variance of the delay between their activations). These values are calculated from the occurrences of learned gestures and actions. Previous studies in HRI have shown that it is important to adjust the delay between human's commands and robots responses to achieve human-like believable behavior.

After completion of the D&A algorithm, the learned acquires a single ABN represented by a Directed Acyclic Graph (DAG) with two node types: $n_i^g$ representing gestures and $n_j^a$ representing actions. Links $l_{g_i a_j}$, exists if and only if gesture $i$ can invoke action $j$. Each link has the mean and variance of the delay between the gesture and corresponding action attached to it. Furthermore, each gesture and action node contains an HMM representing the corresponding gesture or action in $G$ and $A$ respectively. Each one of these HMMs is represented by the tuple $\langle \pi_i, T, B \rangle$ where $\pi_i$ represents the initial state distribution, $t_{ij}$ represents the transition probability from state $i$ to state $j$, and $b_j(k)$ represents the emission probability of output $k$ from state $j$. The mean of all the motifs represented by each node/HMM is also included inside the node. This mean will be used during controller generation.

This representation of the interaction protocol allows the learner to predict the behavior of the actor given the contin-

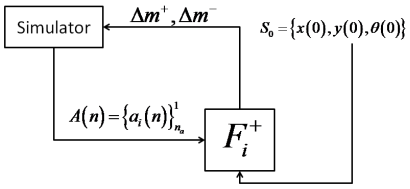



Fig. 3. Learning the Increment function ($F_i^+$) that maps between the current state (action stream) and motor commands during *motor babbling*.

uous $G$ and $A$ streams. The main disadvantage of the system as proposed in this section is that the learner cannot actually *act* in place of the actor because action nodes contain no controllers to activate when the action node is activated. This problem is dealt with in section IV. Another problem is that the whole set of training data is needed to apply the D&A algorithm (batch learning) which disallows the learner from improving its knowledge of the protocol after the ABN was created. Section V deals with this problem.

## IV. Controller Generation

Controller generation is achieved in two sub-stages. Firstly, reinforcement learning is used to allow the learner to generate its basic motion primitives related to the action stream dimensions. This is called the *motion Babbling* stage. Now the mapping between the action stream dimensions and robot's motor space. Once these primitives are learned, the robot starts to generate controllers as sequences of primitives that generate the required pattern in the action dimensions. This is the second sub-stage and is called Piecewise Linear Controller Generation (PLCG).

### A. Motor Babbling

During motor babbling, the robot builds a repertoire of motor primitives related to the action dimensions that will be used in D&A Algorithm. These primitives are constituted from two functions for every dimension: one to increase and the other to decrease that dimension with minimal impact on the value of others. Section IV-B shows how these functions can be used to simplify the development of controllers for the learned actions. To reduce the risks on the learner robot during this phase, a simulator was used. The robot starts in a predefined initial state (specific values for $A$) and tries random motion actions (commands to its motors $C$) keeping track to the changes happening to each of its action dimensions. This training data is then used to learn two functions for every dimension called the increment and decrement functions. These two functions take as input the current action stream state and produce motor commands according to the following rules: The increment function of dimension $i$ (hereafter $F_i^+$) increases the value of this dimension by some rate ($\delta_i$) while keeping the change in other dimensions as small as possible. The decrement function (hereafter $F_i^-$) reduces the value of this dimension by some rate ($\delta_i$) while keeping the change in other dimensions as small as possible.

Thus $F_i^+$ and $F_i^-$ both solve the following constrained optimization (minimization) problem with positive and negative $\delta_i$s:

Objective:

$$\sum_{\substack{j=1:n_a \\ i \neq j}} \left(a_j(n+1) - a_j(n)\right)^2 \tag{1}$$

Constraints:

$$\begin{array}{l} \text{for} 1 \leq i < n_a \\ |a_i(n+1) - a_i(n)| \geq \delta_i^- \\ |a_i(n+1) - a_i(n)| < \delta_i^+ \end{array} \tag{2}$$

Control Variables:

$$C \equiv \left[\Delta m^+(n), \Delta m^-(n)\right] \tag{3}$$

where $a_i(n)$ is the $n$'th sample of the $i$'th action dimension, $\Delta m^+$ is the *sum* of the two commands given to the motors and is proportional to the speed, $\Delta m^-$ is the *difference* between these two commands and is proportional to the rotation angle (differential drive arrangement),$\delta_i^+$ is the upper limit on the required rate of change, and $\delta_i^-$ is the lower limit. The pair $\Delta m^+$ and $\Delta m^-$ constitute the command sent to the motors $C$.

The problem is formulated as a Markov Decision Process (MDP) and is solved using standard Q-Learning with the aid of a simulator that can produce $A(n+1)$ given $A(n)$ and $C(n)$. The resulting $F_i^+$ and $F_i^-$ can now represent a straight line in the action dimension $i$ with an approximate slope of $\delta_i$=$\left(\delta_i^- + \delta_i^+\right)/2$. These two functions thus serve to linearize the relation between the action dimensions and motor commands and in the same time decouples different action dimensions. Fig. 3 shows the outline of learning $F_i^+$. The inputs to the function are not only the current value of the action stream but also the initial state of the robot with respect to the environment.

A particular property of the specific action stream dimensions we selected for this work is that a different slope $\delta_2 = a \times \delta_1$ can be achieved simply by multiplying the final command by the constant $a$. This means that $F_i^+$ and $F_i^-$ need to be learned for a single value for $\delta_i$ and their output can then be scaled to achieve any required slope in the corresponding action dimension. In more complex cases, these functions need to be learned for multiple values of the slope and then interpolated as needed in run-time.

### B. Piecewise Linear Controller Generation

The second and final step in generating the required controllers given the ABN and the $F_i^+$,$F_i^-$ functions. Fig. 4 depicts the proposed approach. Firstly, the mean pattern attached with the node is approximated by a piecewise linear time series using the SWAB algorithm [17]. Secondly, the slope of each line segment is calculated for every action dimension included in the pattern and $F_i^+$,$F_i^-$ functions are used to generate a controller for each of these dimensions. This results in a sequence of $F_i^+$ and $F_i^-$ calls that produce the required pattern.

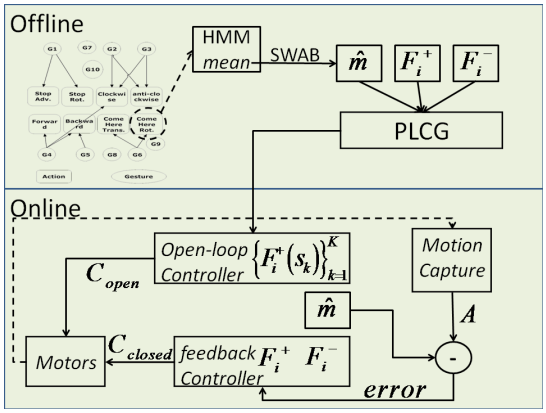



Fig. 4. Piecewise Linear Controller Generation.

This is an open loop controller and in real world can cause large errors. To correct for that, the difference between the actual action stream (state) as perceived by the robot and the piecewise mean approximation ($\hat{m}$) is calculated at a frequency of 10Hz and functions $F_i^+$ and $F_i^-$ are then used to correct for the error with a rate equal to $\pm\zeta\delta_i$ where $\zeta > 0$ and $\delta_i$ is the slope of the current linear segment of $\hat{m}$. In the experiment reported in this paper we used $\zeta = 1$. This closes the loop and produces the final closed loop controller.

This simple approach did work in the navigation case because the motifs in the action stream were mostly piecewise linear. In more complex motif forms (e.g. the gestures in the gesture stream), the applicability of this technique will be limited and further research is required to allow the learner to generate more complex nonlinear controllers.

## V. Accumulation Stage

By the end of the controller generation stage, the learner robot will be able to replace the actor in the guided navigation scenario. Nevertheless, new human partners (operators) will tend to use different gestures or may use some already learned gestures with a different meaning. The robot then needs a mechanism to improve the learned protocol (i.e. the ABN) either by watching new interactions or during its own engagement in guided navigation. This section describes a simple mechanism for achieving this goal.

In this paper, we focus on the situation in which the robot needs to combine two already learned ABNs. After learning an ABN by watching interactions between a new actor/operator pair, the robot needs to combine this ABN with its already existing ABN learned by watching previous pairs of partners to generate a single ABN that captures the gesture/action relations represented by both ABNs. The more general incremental case in which it is required to update a single ABN during interaction will not be discussed in this paper. The evaluation experiment will use only the described two ABN combination approach. The ability to model gesture usage of a single user (without this stage) can still be useful practically if there is a recognition mechanism that allows the robot to recognize with which user it is interacting and adjust the ABN it uses according to this knowledge. Nevertheless, the ability to combine multiple ABNs – provided in this stage – allows the learner to move from *user modeling* to *task modeling* which improves its generalization as will be shown in the evaluation experiment (see section VI). It also allows us to get more insight into the task itself by examining the combined ABN from many users. It allows us also to study the differences between cultures and user groups in interacting with robot by comparing ABNs that were trained with multiple users from different groups.

To combine two different ABNs we need to discover nodes in the two ABNs that represent the same action or gesture and combine them while keeping nodes of every ABN that have no counterparts in the other one. The simplest approach is to compare the HMM parameters stored in each node from the two ABNs (or the mean motif) and combine any two nodes of the same type if the distance between their parameters is less than some predefined threshold. This approach is expected to have limited success because it does not take into account the relation of every node to other nodes in the ABN which prevents it from using the full information embedded in the ABNs in solving the association problem between nodes. In the same time, it is hard to decide the value for the threshold that will give good false rejection/false acceptance balance.

In this paper we try to utilize more information from the two ABNs. The main assumption of the proposed approach is that action nodes will be more similar in the two ABNs than gesture nodes. The justification of this assumption is that action nodes depend on the *task* which is fixed in our case (e.g. guided navigation), while gesture nodes depend on the *human partner* which is much harder to predict as two humans may use very different gestures to mean the same thing (e.g. during our experiments six different gestures were found that mean *stop*). Based on this assumption action nodes first were processed and matched first.

The algorithm starts by compiling two lists of action nodes from the two ABNs (namely $AN^1$ and $AN^2$). For every member of $AN^1$ (called $an_i^1$), the motif mean ($m_i^1$) is compared with every other motif mean in the same ABN ($m_j^1$) using Dynamic Time Wrapping (DTW) and the minimum distance is selected as the similarity threshold of this node $\tau_i$:

$$\tau_i = \min\left(d_{DTW}\left(m_i^1, m_k^1\right)\right) \tag{4}$$

where $1 \leq k \leq n_A^1$, $k \neq i$, and $n_A^1$ is the number of action nodes in $AN^1$.

The second step is to compare the mean of node $i$ with every other node in the second list $AN^2$ and a link $la_{1i}^{2j}$ is created between $an_i^1$ and $an_j^2$ iff: $d_{DTW}\left(m_i^1, m_j^2\right) < \tau_i$ and $\left(d_{DTW}\left(m_i^1, m_j^2\right) - d_{DTW}\left(m_i^1, m_k^2\right)\right) < \eta\tau_i$ for $1 \leq k \leq n_A^2$ and $k \neq j$ for some value of $\eta$ greater than zero. We select $\eta$ to equal 0.25 for all our experiments. If two or more nodes in $AN^2$ satisfy these two conditions, a link is created between $an_i^1$ and each of them. Each link had a value equal to the DTW distance between the means of the two nodes it links.

The third step is to apply the first two steps to all the gesture nodes in the two ABNs (called $GN^1$ and $GN^2$ hereafter). This generates another set of (possibly conflicting) links $lg_{1i}^{2j}$.

The final step is to remove all the conflicts in the two link lists to have at most one node in the second ABN connected to any node in the first ABN. For every link we calculate a *link competence index* (LCI) that evaluates the match between the two ABNs if this link was kept as follows:

$$LCI\left(la_{1j}^{2i}\right) = \frac{1}{v\left(la_{1j}^{2i}\right)} + \lambda_a \sum_{\substack{gn_l^2 \in Par\left(an_i^2\right) \\ gn_k^1 \in Par\left(an_j^1\right)}} LCI\left(\mathrm{lg}_{1k}^{2l}\right) \tag{5}$$

$$LCI\left(\mathrm{lg}_{1j}^{2i}\right) = \frac{1}{v\left(\mathrm{lg}_{1j}^{2i}\right)} + \lambda_g \sum_{\substack{gn_i^2 \in Par\left(an_l^2\right) \\ gn_j^1 \in Par\left(an_k^1\right)}} LCI\left(la_{1k}^{2l}\right) \tag{6}$$

where $Par\,(n)$ is the set of all parents to node $n$. These equations constitute a set of $n_{la}$+$n_{ga}$ equations in the same number of variables and can be solved using a simple iterative approach similar to the value iteration for solving MDPs.

A larger action LCI means that not only the action nodes connected by the link are similar but also their parent gesture nodes are similar as well. A larger gesture LCI means that not only the gesture nodes connected by the link are similar but also their child action nodes are similar as well. The parameters $\lambda_a$ and $\lambda_g$ controls the relative importance of gesture nodes in calculating the LCI of action nodes and vice versa. In our experiments we selected $\lambda_g = \lambda_a = 0.5$.

After the LCI is calculated for every link, the link with highest LCI fanning out from any node is kept and the rest are discarded.

After resolving all conflicts, the nodes in the two ABNs that are still linked are combined and their HMM and motif mean are re-generated from the full set of motif occurrences used when creating the two ABNs.

Combining nodes from two ABNs does not affect the edges except if it caused two nodes to be connected by more than one edge in the final ABN. In this case, the mean and variance of the delay associated with the final edge is calculated from the mean, variance, and number of occurrences in the two combined edges.

The main advantage of this approach is that it utilizes information from the whole ABN in determining the similarity between nodes. Another advantage is that the thresholds are adjusted per node and are determined automatically from the data.

## VI. Evaluation Experiment

This section describes a proof of applicability experiment that was conducted to evaluate the proposed method. 18 subjects were recruited for this experiment (10 males and 8 females) with ages ranging from 18 to 31. All subjects were university students and they had no professional experience in operating robots. 15 of the subjects never operated a robot before. The goal of the experiment was to compare the performance of the learner robot in performing the actor role in guided navigation under three settings:

- WOZ: Wizard of OZ arrangement in which the robot is remotely controlled by a hidden human operator. The hidden operator watches the gestures of the subject and issues motion commands to the robot
- Per-Participant Learner: The robot controller is developed using the first three stages of our approach (without accumulation) from a single interaction with the subject and then used as the actor with the same subject.
- Accumulating Learner: The robot controller is developed using the four stages of our approach and then tested with a subject it never encountered before.

After completing a background evaluation questionnaire and attaching PhaseSpace motion capture markers and B-PACK [15] wireless accelerometers, every subject conducted four sessions. In every session the goal of the subject was to instruct a robot to follow a path drawn in the ground using free hand gestures.

In the first session, the robot was WOZ controlled. This session served two purposes: Firstly, it allowed the learner to collect training data in the form of $G$ and $A$ streams. Secondly, it allowed the subject to get used to controlling the robot using hand gesture. After each session, the subject filled a questionnaire to evaluate the performance of the robot during this session. Because the learning robot did not have any access to the behavior of the WOZ operator (only $G$ and $A$), the learning system is still unsupervised and the WOZ operator can be considered as a part of the actor. The main reason for using this arrangement is not to pre-program the actor with fixed action/gesture relations that can be easily learned.

While the subject was filling the first session questionnaire, the Per-Participant learner applied the first 3 stages of the proposed approach to generate an ABN representing the interaction protocol used by this user.

The accumulating learner combined this ABN with its current ABN but the combined ABN will not be used with this participant but the next one. This ensures that the accumulating learner did not use any information from the user it was tested with in any of the 17 trials (The first participant did not interact with the accumulating learner because there was no ABN available other than the one created using this participant's data).

To test the generalization of the learning model, the path used in the first session during training was different from the path used in the final three test sessions. Fig. 1 shows a snapshot of this experiment and the accompanying video displays a compilation of short clips from it. If the subject was not able to get the robot to the goal within 20 minutes the session was considered a failure and was aborted.

In each post-session questionnaire, the subjects evaluated the robot they interacted with using a scale from 1 to 7 in

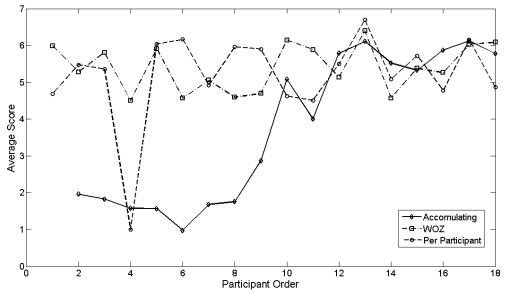



Fig. 5. Average subjective scores assigned by the 18 participants for the three conditions.

the following dimensions:

1) Ease of guiding the robot.
2) Ability of the robot to understand gestures.
3) Attentiveness of the robot.
4) Accuracy of the robot in following the gestures.
5) Naturalness of robot's behavior.

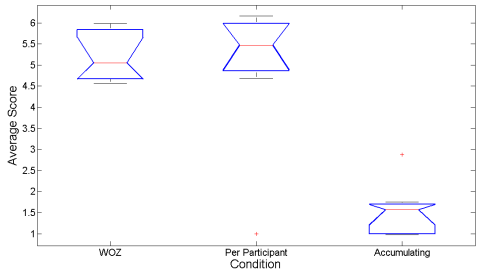
After all the sessions, the subject selected one of the last three robot controllers as her/his preferred actor.

## VII. Results and Discussion

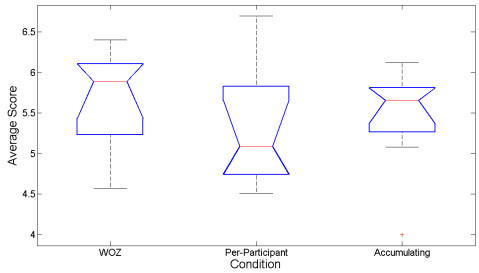
The WOZ condition was always successful. The per-participant learner failed once in the second day, while the accumulating learner failed three times all in the first two days (not including the first participant with which it did not interact because of the unavailability of any accumulated ABN).

Fig. 5 shows the average subjective score assigned by the 18 subjects to the three conditions. As the figure shows, the per-participant learner received a similar average score to the WOZ operated robot from all participants. On the other hand, the first six participants ranked the accumulating learner much less than both the WOZ operator and per-participant learner. One possible explanation of this finding is that the accumulating learner had to rely on training data from participants other than the one with whom it was tested and it needed some time to collect enough gesture types to cover the gestures usually used in this task. During the last two days (last 6 sessions) the accumulating learner caught up with the per-participant learner and the WOZ operated robot and ended in the last session with a score higher than the per-participant learner. An interesting finding from this figure is that the improvement in the accumulating learner's behavior did not happen gradually, but it seems that there is some threshold that the learner passed with the ninth session and its behavior suddenly improved after that. One possible explanation is that the total number of actions in this task is fixed and so the accumulating ABN did not need to include new actions, but the number of gestures that can invoke each of these actions is rather large and once the robot could learn enough of them, it could show a qualitatively different (better) behavior.

To quantify this difference objectively, we calculated the Pearson correlation between the average score in the three




(a) First Nine Participants




(b) Last Nine Participants

Fig. 6. The change of the scores assigned by participants to the three conditions over time.

conditions and the participant number. The accumulating learner showed a correlation of 81.94% with the participant order and the p-value was 0.000032 showing a statistically significant correlation. The WOZ and per-participant learner showed no statistically significant correlation with the participant order.

Fig. 6-a shows the mean and standard deviation of the average scores assigned by the first nine subjects to the three conditions. The t-test was used to check the significance in mean differences shown in the figure. In the case of the WOZ operated robot and the per-participant learner, the difference in mean was not statistically significant (p-value=0.4653). In the case of the accumulating learner, the difference was statistically significant between it and the WOZ operated robot (p-value$<$0.001) and the per-participant learner (p-value=0.0052). Wilcoxon rank sum test also agreed to this result

Fig. 6-b shows the mean and standard deviation of the average scores assigned by the last nine subjects to the three conditions. Again, using t-test and Wilcoxon rank sum test, no differences were found to be statistically significant in this case.

This fact that the behavior of the per-participant learner and the WOZ operator was similar in all cases suggests that the first three stages of the proposed approach were successful in generating the needed controllers from a single training session.

The high correlation between the average score of the accumulating learner and the participant order suggests that the accumulating learner did improve its behavior over time and supports the claim that the ABN combination approach described in section V was successful at least in this task

and this experimental settings.

The participants selected the WOZ operated robot as their preferred robot eight times, the per-participant learner seven times and the accumulating learner three times (all in the last two days). Again, the effect of time on the evaluation of the accumulating learner was evident. In the first two days no one preferred the accumulating learner, while in the last two days half of the participants preferred it even over the WOZ operated robot. The WOZ operated robot and per-participant learner were each preferred three times in the first two days and the per-participant learner was preferred two times (compared to one for the WOZ operated robot) in the last two days. These findings confirm the results found from analyzing the post-questionnaire scores. Due to lack of space the objective evaluation based on completion time and accuracy will not be presented here but they confirm the findings of the subjective evaluations presented in this section.

## VIII. Limitations and Future Work

The proposed approach was designed to allow the learner to *copy* the behavior it perceives from the actor in an unsupervised way. Extension of the learned behavior to other tasks was not considered in this work.

Another limitation of the proposed approach is the assumption implicit in the motor babbling algorithm that the primitives needed in the action dimensions are linearizable and that it is possible to decouple these dimensions for easier control during the PLCG phase. This assumption was true for the guided navigation task and the specific action stream signals we selected, but there is no guarantee that the proposed $F_i^+$ and $F_i^-$ learning mechanism will converge in other cases. In the future automatic generation of controllers when the action stream is coupled will be addressed.

The final limitation we discuss here is that the system was designed assuming no explicit feedback from the actor to the operator. Again this was acceptable in the guided navigation settings but to generalize the approach to more teammate situations, a third stream of feedback signals needs to be added to the discovery phase with corresponding controller generation. The proposed approach should be easily extended in this manner to take care of simple interactions with feedback but more research will be required to extend it to more complex human like spontaneous interactions.

## IX. Conclusion

This paper presented a new approach to unsupervisedly learn simple interaction protocols in the form of an Augmented Baysian Network and automatically generating the required controllers to actually participate in the learned interactions. The approach was successfully applied to learn guided navigation using free hand gestures with no assumptions about the actions related to the task, the number of gestures used, their durations, or their occurrence patterns. The proposed approach also allows the learner to combine different learned ABNs to improve its performance over time and to accommodate different gestures used by different human partners.

The paper also reported a proof of applicability experiment using 18 untrained users who conducted 72 guided navigation sessions with a cart robot. The results of this experiment show that the proposed approach was successful in allowing the learner robot to achieve undistinguishable behavior from a WOZ operated robot after a single training session. It also showed that the ABN combination algorithm allows the learner to improve its performance over time and allows it to interact with new participants without any need for re-training.

## References


[1] T. Hashiyama, K. Sada, M. Iwata, and S. Tano, "Controlling an entertainment robot through intuitive gestures," in *2006 IEEE International Conference on Systems, Man, and Cybernetics*, 2006, pp. 1909–1914.

[2] B. D. Argall, S. Chernovab, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, 2009.

[3] Y. F. O. Mohammad and T. Nishida, "Eica: Embodied interactive control architecture," in *Twentieth Australian Joint Conference on Artificial Intelligence*, 2007, pp. 357–366.

[4] C. A. A. Calderon and H. Hu, "Robot imitation from human body movements," in *In Proceeding AISB05 Third International Symposium on Imitation in Animals and Artifacts*, 2005.

[5] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, pp. 3519–3524 Vol.4, 26-May 1, 2004.

[6] ——, "Grasp recognition for programming by demonstration," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 748–753, April 2005.

[7] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[8] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions: Biological Sciences*, no. 1431, pp. 537–547.

[9] T. Ogata, S. Sugano, and J. Tani, "Open-end human robot interaction from the dynamical systems perspective: mutual adaptation and incremental learning," in *IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*. Springer Springer Verlag Inc, 2004, pp. 435–444.

[10] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 425–431, Dec. 2006.

[11] H. Kadone and Y. Nakamura, "Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3548–3553, Aug. 2005.

[12] D. Kulic, W. Takano, and Y. Nakamura, "Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains," *The International Journal of Robotics Research*, pp. 761–784, 2008.

[13] Y. F. O. Mohammad, T. Nishida, and S. Okada, "Unsupervised simultaneous learning of gestures, actions and their associations for human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2009, pp. 2537–2544.

[14] Y. Mohammad and T. Nishida, "Constrained motif discovery in time series," *New Generation Computing*, no. 27, pp. 319–346, 2009.

[15] R. Ohmura, F. Naya, H. Noma, and K. Kogure, "B-pack: a bluetooth-based wearable sensing device for nursing activity recognition," *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 6–10, Jan. 2006.

[16] Www.phasespace.com.

[17] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, 2001, pp. 289–296. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2001.989531