

# Learning to Understand Parameterized Commands through a Human-Robot Training Task

Anja Austermann, Seiji Yamada

**Abstract**— We propose a method to enable a robot to learn simple, parameterized commands, such as “Please switch on the TV!” or “Can you bring me a coffee?” for human-robot interaction. The robot learns through natural interaction with a user in a special training task. The goal of the training phase is to allow the user to give commands to a robot in his preferred way instead of learning predefined commands from a handbook. Learning is done in two successive steps. First the robot learns object names. Then it uses the known object names to learn parameterized command patterns and determine the position of parameters in a spoken command. The algorithm uses a combination of Hidden Markov Models and Classical Conditioning to handle alternative ways to utter the same command and integrate information from different modalities.

## I. INTRODUCTION

WHEN creating robots, that can interact with non-experts in everyday tasks, one of the challenges is to enable the robot to understand commands given by its user in a natural way. This paper describes an ongoing study that attempts at solving this problem by making the robot learn simple parameterized commands and feedback through natural interaction with a user.

We have already proposed a technique [2] for learning to understand positive and negative feedback through human-robot interaction. In this paper this method is extended to deal with more complex, parameterized utterances. While positive and negative feedback utterances do not need to be segmented but can be processed as a whole, commands may contain different parameters, which need to be handled by the system. For example, the command “Put the *book* on the *table*!” contains an object name and a place name. In order to understand the meaning of the whole utterance, the command and its parameters need to be segmented. Our system learns so-called “command patterns”. That is, it does not try to analyze the grammatical structure of a command, but rather uses placeholders for the parameters and models the rest of the command as a whole. This is less flexible than a real grammatical analysis but can be used more easily to model a user’s typical ways of uttering commands.

A lot of research has been done on automatic symbol grounding for robots [3][4][9]. Symbol grounding is a

Manuscript received July, 2009.

Anja Austermann is with the Graduate University for Advanced Studies (SOKENDAI), Tokyo, 101-8430 JAPAN (e-mail: anja@nii.ac.jp).

Seiji Yamada, is with the National Institute of Informatics and the Graduate University for Advanced Studies (SOKENDAI), 101-8430 JAPAN (e-mail: seiji@nii.ac.jp).



Fig. 1: Aibo performing Training Task

complex task in which symbols, such as the words of a natural language, are connected with meanings, that is objects, places, actions etc. in the real world. It often involves visual recognition and naming of objects or actions. Our work has a slightly different focus. We concentrate on learning how a certain user utters commands and feedback, but assume that the robot already knows basic symbolic representations of the actions, that it is able to perform and the objects/places, it can recognize, like  $move(objectA, placeB)$ . In order to react to natural, multimodal commands and feedback, it needs to learn a mapping between these existing symbolic representations and commands, object/place names or feedback given naturally using speech, prosody and touch. This enables the robot to deal with instruction given by the user in his or her preferred way. Assuming, that basic grounded symbols already exist by the time of the training is a quite strong requirement, but this is likely to be the case for typical service- or entertainment robots as they normally have a set of built-in functions and can visually recognize and manipulate certain objects in their environment.

In this paper we propose a combination of special training tasks, which allow a robot to provoke commands and feedback from a user, and a two-staged learning algorithm,

which has been designed to resemble the processes, which occur in human associative learning.

In a real-world scenario, the training tasks, which allow the robot to adapt to its user, would have to be performed before actually starting to use the robot. In order to allow for a quick and easy training, for example in front of the TV/PC screen, we use “virtual” training tasks. For our experiments on command learning we created an animated virtual living room. It is a simplified 3D-model of a living room, which can be seen in Fig. 1 and Fig. 2. The virtual living room is projected on a white screen and the robot uses motions, sounds and its LEDs to show which moves it is making. Appropriate animations are shown in the virtual living room for each move.

In order to learn the correct meaning of its user’s utterances, the robot needs to know in advance, which commands the user is going to utter. This is ensured by the design of the training task. The user is informed about which actions he should make the robot perform, by typical and easily recoverable changes in the living room, such as a carpet getting dirty or a book falling from the shelf. Moreover, the system can display thought balloons representing desires of the user like wanting to drink a coffee or wanting to know the battery state of the robot. Details on the tasks are given in section III.

Having a robot learn commands from its user instead of forcing the user to learn commands to control the robot, has different advantages. As it shifts the learning effort from the user to the robot, it would be especially desirable for elderly people with memory deficits to have a robot adapt to their natural way of giving commands and feedback to it.

## II. RELATED WORK

There are various approaches towards symbol grounding and learning to understand spoken utterances, especially names of objects or actions and connect them with their visual representations.

Roy [8] proposed a model of cross-channel early lexical learning to segment speech and learn the names of objects, which are recorded by a camera. He used models of long term memory and short time memory to find and learn recurring auditory patterns, which are likely to be object names. He used insights from infant-word learning and recorded the speech samples for training the robot through experiments with mothers playing with their infants.

Iwahashi [4] described a method to learn to understand spoken references to visually observed objects, actions and commands which are a combination of objects and actions. In a second stage, the robot learned to execute the appropriate actions that have been demonstrated by the instructor before, in response to commands from its instructor. Iwahashi applied Hidden Markov Models to learn verbal representations of objects and motions perceived by a 3D-camera.

Steels and Kaplan [10] developed a system to teach the names of three different objects to an AIBO pet robot. They used so-called “language games” for teaching the connection between visual perceptions of an object and the name of the

object to a robot through social learning with a human instructor.

In [1] and [2] we outlined an approach to enable a robot to learn positive and negative feedback from a user through a training task. We reached an average accuracy of 95.97% for the recognition of positive and negative reward based on speech, prosody and touch. The current work is an extension of this approach to allow the system to deal with parameterized commands. At the moment, we do not use actual vision processing but use virtual training tasks, which allow the robot to access all features of the task directly without additional processing. Learning to understand commands through virtual training tasks, instead of teaching them, for example, by demonstration has two main advantages. It enables the robot to learn commands, which would be difficult to teach by demonstration, such as asking the robot about its battery status or telling it to switch itself off.

The training tasks also allow the robot to take over the active role in the learning process by requesting specific learning tasks for certain objects/places or commands from the task server. This enables the robot to systematically repeat the training of feedbacks, commands or object/place names that have not received sufficient training, yet.

By combining Hidden Markov Models and classical conditioning, our algorithm can handle multiple ways to utter the same command and integrate information from different modalities.

## III. TRAINING TASK

The robot learns to understand the user’s commands and feedback in a training phase. The design of the training phase is a key point for our learning method because it enables the robot to provoke commands as well as feedback from the user. For training the robot, we use computer-based “virtual” training tasks. We implemented a *virtual living room* which shows a simplified 3D model of a living room. It is shown in Fig. 2. Virtual training tasks allow the robot to immediately access all properties of the task, such as the locations of objects etc. through a connection to the task server. Moreover, virtual tasks can be solved without time-consuming walking or



Fig. 2: Virtual Living Room.

other physical actions, which cannot be performed by the AIBO, such as actually cleaning or moving around different objects. This is important for our experiments.

We have implemented a framework, which can easily be extended to fit different tasks, robots or virtual characters. The virtual living room that we use for our experiments is projected to a white screen and the robot uses motions, sounds and its LEDs to show which move it is performing (Fig. 1). During the training the robot cannot actually understand its user but needs to react appropriately to ensure natural interaction. This is done by designing the training task in a way that the robot can anticipate the user’s commands.

During the training phase, the robot sends the requests, which object, place or command and reward it wants to learn to the task server. The task server then visualizes the expected command or highlights the requested object/place on the screen in a way that the user can understand it easily. It also sends relevant information, such as the coordinates of objects back to the robot, so that it can, for example, perform a pointing gesture to ask for an object or place name. When the user utters a command, the robot can either perform a correct or incorrect action to provoke positive or negative feedback from the user. This way, the robot is able to explore the user’s way of giving different commands as well as feedback.

The system can only learn verbal representations of simple commands consisting of one action and the related objects. Table 1 shows the set of commands that the robot learns in our experiments along with their parameter signature and an example of a sentence that the user might utter.

TABLE 1: COMMAND NAMES AND PARAMETERS

Command	Parameters	Example sentence
move	object, place	Put the ball into the box.
bring	object	Bring me a coffee, please.
open	object	Hey AIBO, open the door.
close	object	Can you close the window?
clean	object	Please clean up the carpet.
switch on	object	AIBO, switch on the light.
switch off	Object	Switch off the radio.
charge battery	<none>	Recharge your battery.
shutdown	<none>	Go to sleep.
show status	<none>	What is your status?
stand up	<none>	Stand up, please.
sit down	<none>	Sit down.

The robot first learns names of objects and places, which can then be used as parameters when learning command patterns. When enough object names are known, the robot continues with learning command patterns like “switch the <object> on!”, “Please move <object> to <place>” etc.

In order to enable the robot to learn, the system needs to make the user give commands in his preferred way but with a predefined meaning. This is done by showing situations in the virtual living room, where it is obvious which task needs to be performed by the robot. Thought balloons with appropriate

icons are used to visualize desires of the user, which cannot be understood easily from the state of the virtual living room alone, such as wanting a coffee or wanting the robot to shutdown. Text is not used in order to avoid any influence on the wording of the user. Some examples of command visualizations and possible commands from the user are:

- It is getting dark and the light is still switched off  
*Switch the light on!*
- A dirty spot on the carpet  
*Clean the carpet, please!*
- A book has fallen off the shelf  
*Can you put the book on the shelf?*
- An icon showing a battery and a question mark?  
*What is your battery status?*
- A thought balloon showing a battery and a connector  
*Go to your charging station!*

#### IV. LEARNING METHOD

The learning algorithm is divided into a stimulus encoding phase and an associative learning phase. This is modeled after natural learning in humans and animals. In the stimulus encoding phase, the system trains Hidden Markov Models (HMMs) to model command patterns, object/names which are used as parameters, as well as positive and negative rewards based on speech, prosody and touch stimuli from the user.

In the associative learning phase, the system associates the trained models with a known symbolic representation, integrating the data from different modalities. For example, it associates an HMM of the utterance “Could you please move <A> to <B>” with the known symbolic representation *move(object, place)* or the utterance “Good robot” and a touch of the head sensor with positive reward. An example of a data structure resulting from this learning process is shown in Fig. 3. The representation of place and object names is not shown in the figure. It can be found in Fig. 4.

##### A. Stimulus Encoding

In the stimulus encoding phase the system trains models of its user’s feedback, commands, and object/place references. The learning is based on Hidden Markov Models for speech as well as for prosody and a simple duration-based model for touch. For each command or feedback, given by the user, the

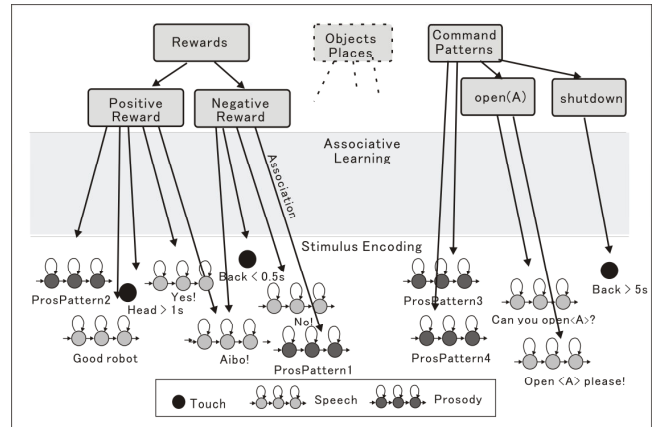


Fig. 3: Example of the Data Structure after Learning.

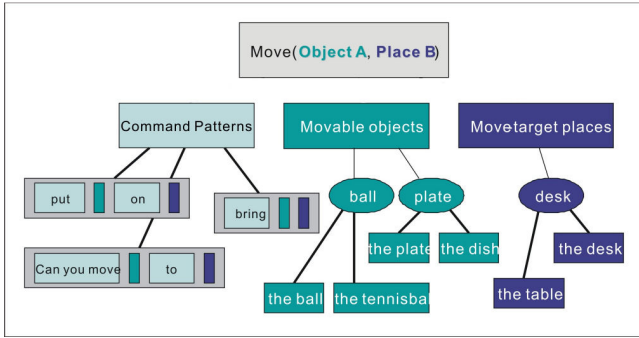


Fig. 4: Command Data Structure.

best matching speech, prosody and touch models are determined according to the methods, described in the following paragraphs. If there is no good existing model, a new one is created. Otherwise, the best matching model is retrained with the data corresponding to the observed stimulus. When retraining has finished, the models are passed on to the association learning stage.

### 1) Speech

For learning commands, we assume that speech is the most important modality. We distinguish three different kinds of utterances, that the speech stimuli encoding needs to deal with: positive/negative feedback, names of objects/places and command-patterns. Command-patterns can have a variable number of slots for inserting object- or place-names like “Stand up”, “Clean <object> please” or “Can you move <object> to <place>?”. An example of a command structure is shown in Fig. 4. The leaves of the tree are trained HMMs. The inner nodes are symbolic representations of objects and command patterns. The thick lines represent associations, learned later in the associative learning phase.

Feedback-utterances, names of objects/places and commands without any parameters can be trained as single HMMs. In case of commands with one or more parameters, the system needs to model the corresponding command pattern using multiple HMMs to allow the insertion of HMMs representing objects/places used as parameters, as shown in Fig. 4. In order to learn a command pattern consisting of multiple HMMs, the system must first determine which parts of the utterance belong to the verb pattern itself and which parts belong to its parameters. From the training task, the system knows which parameters to expect. The algorithm uses this information to locate object/place names in the utterance by matching the utterance against all HMMs that have an existing association to the expected parameters. To do so, a grammar for the recognizer is generated automatically from the already trained object names. In case of a command with two parameters, object1 and object2, the grammar looks as follows:

$Object\_1 = Utterance1 \mid Utterance2 \mid Utterance3 \dots$   
 $Object\_2 = Utterance4 \mid Utterance5 \dots$

Searchstring =  
 $([Sil] [Garbage] Object\_1 [Garbage]$   
 $Object\_2 [Garbage] [Sil] ) \mid$

$([Sil] [Garbage] Object\_2 [Garbage]$   
 $Object\_1 [Garbage] [Sil] )$

The utterances 1 to 5 in this grammar are all utterances that have an association to either object 1 or object 2. The garbage model is trained with all utterances of the speaker. The silence model is trained with only background noise. Matching is done using HVite, an implementation of the Viterbi algorithm in the Hidden Markov Model Toolkit (HTK) [11]. Running the recognizer with this grammar returns the positions of the parameters in the utterance. The utterance is then cut at the boundaries of the detected parameters. All parts that do not belong to the name of an object or place are expected to belong to the command pattern and used to create or retrain HMMs. The places, where object- or place-names have been cut out are modeled as slots in the grammar of the utterance recognizer.

To model speech utterances our system trains one user-dependent set of utterance HMMs for each of object/place names and feedback, and a set of HMM-sequences for learning command patterns. As a basis for creating these utterance models the system uses an existing set of monophone HMMs. It contains all Japanese monophones and is taken from the Julius Speech Recognition project [5].

As the robot learns automatically through interaction, no transcription of the utterances is available. Therefore, an unsupervised clustering of perceived feedbacks that are likely to correspond to the same utterance is necessary. The system solves this problem by using two recognizers in parallel: One recognizer tries to model the observed utterance as an arbitrary sequence of phonemes. The other recognizer uses the feedback, object/place or command models, trained so far, to calculate the best-matching known utterance.

In case of command patterns each of the parts before, between and after parameters is modeled as a separate HMM/phoneme sequence as shown in Fig. 4. An appropriate

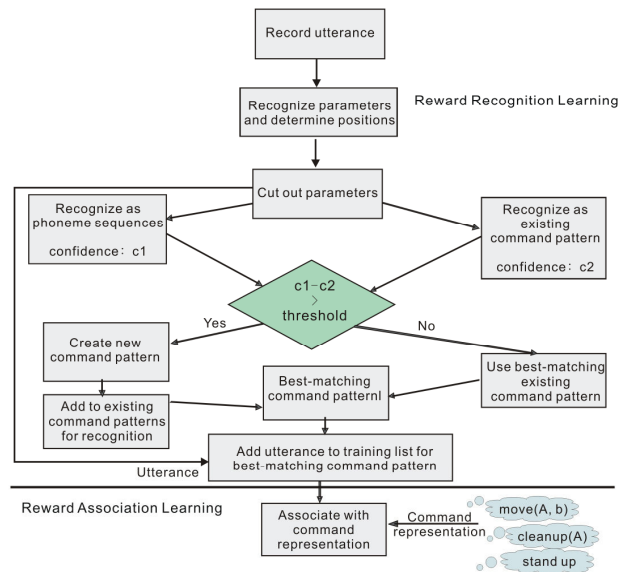


Fig. 5: Control Flow for Learning Command Patterns.

recognition grammar is used to keep together the parts that belong to one command.

Every time an utterance from the user is observed, first the system tries to recognize it with both recognizers. Recognition is done by HVite [11]. The recognizers return the best-matching phoneme sequence and the best matching model of the complete feedback, object name or command pattern. Moreover, confidence levels are output for both recognition results. The confidence levels, which show the log likelihoods per frame of both results, are compared to determine whether to generate a new model or retrain an existing one.

In case of an unknown utterance, the phoneme-sequence based recognizer typically returns a result with a noticeably higher confidence, than the one of the best matching utterance model. For a known utterance, the confidence corresponding to the best-matching utterance model is either higher or similar to the best-matching phoneme-sequence. Therefore, if the confidence level of the best-fitting phoneme sequence is worse than the confidence level of the best-fitting utterance model or less than a threshold better, then the best-fitting utterance model is retrained with the new utterance. The threshold is determined experimentally from the speech data recorded in the experiment. In case of command patterns each of the HMMs modeling a part of the command pattern is retrained separately with the corresponding part of the utterance, which has been determined in the first step.

If the confidence level of the best-matching phoneme sequence is more than a threshold better than the one of the best-fitting whole-utterance model, then a new utterance model is initialized for the utterance. The new model is created by concatenating the HMMs of the recognized most likely phoneme sequence to a new HMM. In case of command patterns one HMM is created for each part before, in between and after the slots for inserting parameters and a grammar defines the order of the individual parts as well as the positions of the parameters. The new model is retrained with the just observed utterance and added to the HMM-set of the whole-utterance recognizer. So it can be reused when a similar utterance is observed. An overview of the training for learning a command pattern is shown in Fig.5.

During the training phase, utterances from the user are detected by a voice activity detection based on energy and periodicity of the perceived audio signal.

### 2) Prosody

We have implemented the recognition of prosody mainly to enhance the learning and recognition of positive and negative feedback. As we do not assume, that prosody can be effectively used to discriminate between different commands or object names, we decided to use only three classes for the prosody based recognizer: positive reward, negative reward and commands. This can be seen in Fig. 3. While speech and touch stimuli are associated with individual commands, prosody only discriminates between these three categories.

For the prosody recognition, utterances are always processed as a whole without locating and cutting out parameters. The HMMs that we use for interpreting prosody are based on features [6] extracted from the speech signal. In order to obtain these features, the signal is first divided into frames of 32 ms length with 16 ms overlap. For each frame, the system calculates a feature vector containing the pitch, the pitch difference to the previous frame, the energy, the energy difference to the previous frame and the energy in frequency bands 1-n. The sequence of feature vectors is used for training the HMMs.

Additionally, the algorithm calculates some global information based on all frames belonging to one utterance. These are the average, minimum and maximum, range and standard deviation and the average difference between two frames for pitch as well as energy. For determining, which HMM is trained with which utterances, the system uses these global features. Utterances with similar global features are clustered and one HMM is trained for each cluster.

### 3) Touch

The user can also interact with the robot using its touch sensors on the head and on the back. We assume that touch is more important for learning rewards than for learning commands. However, we want to give the users the possibility to use touch to express commands: e.g. use a long press of the back touch sensor to put the robot into sleep mode. As we do not assume, that users will use touch to encode names of object or places, no associations are learned between touch patterns and objects/places.

To encode touch, we use its duration and whether the head or the back sensor was touched. We use three categories for short ( $< 0.5$  s), medium ( $0.5s < x < 1$  s) and long ( $> 1$  s) touches.

For learning to understand positive or negative feedback, we did not take into account the exact sequence of short, medium and long touches in our previous approach. However, if the user employs touch to encode commands, the exact sequence may be important. The observed sequences of short, medium and long touches representing a command or feedback are encoded as strings, such as "LB,SH,LH" for a long touch of the back sensor, a short touch of the head sensor and a long touch of the head sensor.

A table is used to store all known touch patterns. For each observed command or feedback the system tries to find the pattern in the table and creates a new entry if necessary. The entry number is then passed on to the associative learning stage.

### B. Associative Learning

We use classical conditioning to establish associations between the known symbolic representations of actions, rewards and objects/places and the trained HMMs for command patterns and parameters. As in our previous approach to learning positive and negative rewards [2], we

employ the Rescorla-Wagner model [7] to learn and update the associations.

The symbolic representations of feedback, commands and their parameters are used as unconditioned stimuli. The HMMs, encoding stimuli coming from the user, are used as conditioned stimuli. The three different kinds of stimuli - feedback, command patterns and parameters - are handled separately from each other.

For speech, associations to HMMs are learned for the symbolic representations of feedback, of objects/places and for the different commands. For prosody, associations are learned toward either positive or negative feedback or the symbol “command”, which stands for any command. This way, prosody helps to distinguish between feedback and commands from the user. Touch models can be associated with positive or negative feedback as well as with different command patterns, but not with objects/places, as we do not assume, that users encode object or place descriptions into touch patterns.

Classical conditioning has different desired properties, such as blocking, secondary conditioning and sensory preconditioning which allow the system to integrate and weight stimuli from different modalities, emphasize salient stimuli and establish connections between multimodal conditioned stimuli, e.g. between certain utterances and touches or prosody patterns.

## V. EXPERIMENTS

We are currently conducting experiments to evaluate the performance of our learning method. The experimental setting is shown in Fig. 6. The system records speech using a close-talk microphone. Video is recorded for later integration of gesture recognition.

The participants are instructed to teach the robot in two phases. In the first phase, they teach object- and place names to the robot. After the object learning has finished, the experiment continues with the teaching of commands. The users are instructed to utter commands, which match the situation shown in the “virtual living room” scene and give positive or negative feedback depending on whether the robot has reacted correctly or not.

## VI. DISCUSSION

We proposed an approach to learn parameterized commands for human-robot interaction. The main restriction of our approach is that it is only applicable as long as the number of commands that the robot needs to understand does not grow too large. Otherwise, learning commands would probably be too time-consuming for real-world use. The learning of object names with our approach can continue after the training phase in a real environment provided the robot can visually identify objects. However, the learning of commands heavily relies on the virtual training tasks to make the user utter the commands that the robot wants to learn.

At the moment, the system can only deal with names of

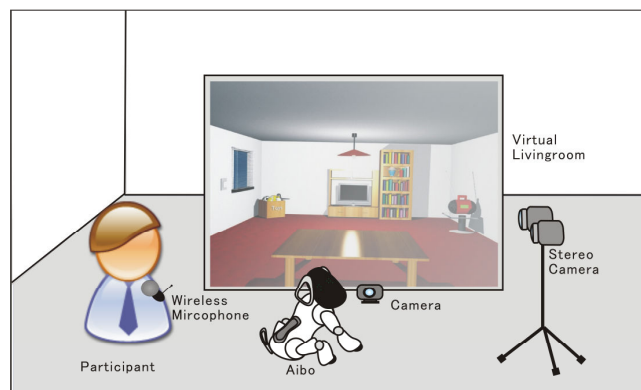


Fig. 6: Experimental Setting.

objects or places, not with descriptions. “The blue cup” or “the cup on the table” would be learned as one object name. In order to allow for more flexible instructions from the user, it is necessary to extend our learning method to enable the system to learn prepositions and certain attributes, such as colors, which are commonly used to distinguish different objects of the same class. Pointing gestures are also frequently used to disambiguate or even replace spoken object references. Therefore, integrating basic pointing gesture recognition is one of the priorities of our ongoing work.

## REFERENCES

- [1] A. Austermann, S. Yamada, “A biologically inspired approach to learning multimodal commands and feedback for Human-Robot Interaction”, CHI Work-In-Progress, 2008
- [2] A. Austermann, S. Yamada, “Teaching a Pet Robot through Virtual Games”, Proceedings of the IVA '08, pp. 308 – 321, 2008
- [3] X. He, T. Ogura, A. Satou, O. Hasegawa, “Developmental Word Acquisition and Grammar Learning by Humanoid Robots Through A Self-Organizing Incremental Neural Network”, IEEE Transactions on Systems, Man and Cybernetics, 37 (5) pp. 1357-1372, 2007.
- [4] N. Iwahashi, “Robots that Learn Language: A Developmental Approach to Situated Human-Robot Conversation”, in Sanker, N. ed. Human-Robot Interaction, pp.95-118. I-Tech Education and Publishing, 2007.
- [5] The Julius Speech Recognition Project: <http://julius.sourceforge.jp>
- [6] T. L. Nwe, S. Foo, S. Wei; L. De Silva, “Speech emotion recognition. using hidden Markov models”, Speech communication 41,4, 2003
- [7] R. Rescorla, A. Wagner, “A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement.”, Classical Conditioning II: Current Research and Theory (Eds Black AH, Prokasy WF) New York: Appleton Century Crofts, pp. 64-99, 1972
- [8] D. Roy, “Grounded Spoken Language Acquisition: Experiments in Word Learning”. IEEE Transactions on Multimedia, 5(2) pp.197-209, 2003
- [9] L. Steels, “Evolving Grounded Communication for Robots”, Trends in Cognitive Science, 7 (7), pp. 308-312, 2003
- [10] L. Steels and F. Kaplan, “AIBO's first words : The social learning of language and meaning”, Evolution of Communication, 4(1) pp. 3-32 , 2001
- [11] S. Young et al., "The HTK Book" HTK Version 3, 2006 <http://htk.eng.cam.ac.uk/>