Robot Learning Rules of Games by Extraction of Intrinsic Properties

Grégoire POINTEAU, Maxime PETIT, Peter Ford DOMINEY
Robot Cognition Laboratory
INSERM Stem Cell and Brain Research Institute
Lyon, France
gregoire.pointeau@inserm.fr, maxime.petit@inserm.fr, peter.dominey@inserm.fr

***Abstract*—A major open problem in human-robot interaction remains: how can robots learn from non-technical humans? Such learning requires that the robot can observe behavior and extract the sine qua non conditions for when particular actions can be produced. The observed behavior can be either the robots own explorative behavior, or the behavior of humans that it observes. In either case, the only additional information should be from the human, stating whether the observed behavior is legal or not. Such learning may mimic the way that infants learn, through interaction with their caregivers. In the current research we implement a learning capability based on these principals of extracting rules from observed behavior using "Human-Robot" interaction or "Human-Human" interaction. We test the system using three games: In the first, the robot must copy a pattern formed by the human; in the second the robot must perform the mirror action of the human. In the third game, the robot must learn the legal moves of Tic Tac Toe. Interestingly, while the robot can learn these rules, it does not necessarily learn the rules of strategy, which likely require additional learning mechanisms.**

***Index Terms*—learning machine; robotics; iCub; *Reactable*; human-robot interaction; human feedback**

I. INTRODUCTION

How is a robot able to understand a new game just by watching it or by playing it without any prior assumption ? How can he differentiate two different games without explanation only by the simplest feedback : a simple "yes" or "no" from a teacher ? What is the limit of simple imitation [1] ? The answer lies in the understanding of the actions [2]: the intrinsic properties of each move. And from this properties, the extraction of *sine qua none* conditions. In our studies, we want to teach a robot to play different board games (as a Tic-Tac-Toe for example) without giving it any assumption about the game. The idea of learning a new concept as it has been done in Wu in [3] but here, we will not use any Bayesian statistics but only simple deterministic algorithm. Breazeal et al. have shown the efficiency of socially guided exploration ([4] and [5]) and social interaction also has been shown crucial in learning ([6] and [7]). Most board games use a set of "natural" rules as the turn taking, that we give to the robot. In our case, we use an humanoid robot called *iCub*, in interaction with an Human through an interactive table [8]: the *Reactable* [9]. Only with a feedback : *"yes or no"* from the user, for each move (is the move is legal or not), the robot is able to discriminate different games and to learn a new game. Also, the robot can learn a game, just by watching 2 humans playing. In this paper we will first see the global architecture and overview of the system, then the experimental protocol and the games used, then the learning part directly, and finally the results of simulated "Human-Robot" interactions and "Human-Human" interactions.

II. SYSTEM DESIGN OVERVIEW

A. Robotic Platform : iCub

The iCub [10] is a humanoid robot open-source platform. It has 53 actuated degrees of freedom (with 19 for each 5-fingers hand) and with a height of 104 cm, has a morphology approximating that of a 3 year-old child.

 ISBN: 978-1-61208-250-9

The distributed modules used to run the robot are interconnected with the open source library YARP [11] through "ports". The ports can exchange data over different network protocols such as TCP and UDP.

The motor control is managed by a Passive Motion Paradigm approach [12], where hand trajectories for reaching actions are computed according to virtual force fields with attractor (target) and repellor (obstacles).

B. Sensor System : Reactable

In the current research we extend the perceptual capabilities of the iCub with the *Reactable*. This is a tabletop tangible interface [9] licensed by *Reactable* Systems. This allows the human and the iCub to manipulate objects in a shared space that the iCub can perceive with high precision. An infra-red illumination beneath the table allows reacTIVision [13], a detection system based on an infrared camera, to accurately and in real-time identify and track tagged object (using fiducial markers [14]) placed on the translucid table. It can also recognize fingertips (cursor) which allow the user to manipulate the digital information (music, game, ...) with real-world objects or direct tactile control (see Figure. 1).

C. Knowledge Base : OPC

The different knowledge of the robot is centralized in a database called OPC for ObjectPropertiesCollector, and grouped accordingly into some enitites : the informations provide in particular by the *Reactable* (e.g. position x and y of an object with the *Reactable* id i) are merged with some ground knowledge (e.g. the object with *Reactable* id x is called "$cross$" and has a size of x,y,z) or reasoning conclusion (e.g. "$onTable1$" if an object is placed on the table) in order to have a full picture about this object. These entities can be accessed with unique identifiers and are managed dynamically in real-time.

The robot knowledge is initialized through the OPC set-up. The database includes :

- locations : 3x3 squares (from A-1 to C-3) to obtain a board game setup, and two other places to put respectively the cross and circle object when unused.
- objects : Two kind of pawns, cross in blue and circle in red, and a special object, the eraser to undo a move if needed.

D. Spoken Language and Supervisor : CSLU-RAD

The spoken language interaction is implemented with the CSLU Toolkit [15] Rapid Application Development (RAD) under the form of a finite-state dialogue system where the speech synthesis is done by Festival and the recognition by Sphinx-II. In addition to provide a human-robot language-based interaction during games, it is used in particular to extract oral feedback from the user about the move quality done by the robot.

This state machine is built in what we called the Supervisor. It has two main functions : guide the human into the games, inviting him to act or choose between options, and to control the robot behavior (including speech of the iCub) using human spoken feedback, the knowledge base or some external module, like BoardGameLearning.

E. BoardGameLearning

BoardGameLearning is a module developped in c++, responsible for the statistical analysis and the learning. We will explain how it works in paragraph IV.

III. Experimental Protocols

A. Games used

For our studies, we use 3 different games, with different properties that we wanted to test with only one learning module. All of them work with a turn taking managed by the supervisor (see II-D and Figure 2).

 ISBN: 978-1-61208-250-9

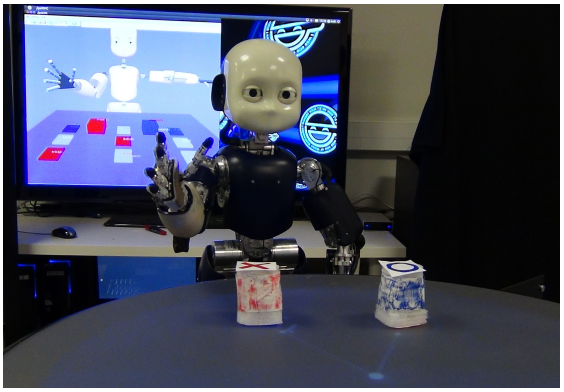

Fig. 1. Normal functioning of the system with iCub and objects on the *Reactable*. A screen behind the iCub displays the iCub's internal representation of the environment including the board game on the *Reactable*. The iCub can manipulate the two stamp for marking of cross or circle in the game. The Reactable detect the position of the objects.

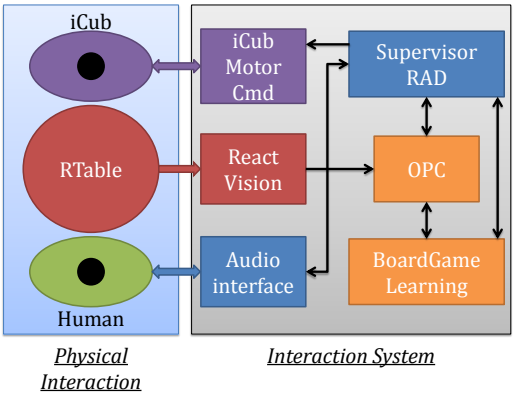

Fig. 2. System architecture. In purple is the robot-related. In blue is the supervisor and interaction related. In orange in the computation and learning module part. In red, the Reactable.

1) Pattern game: The first game is a Pattern Game. The iCub and the Human, with alternating turn taking, have to fill the board with a predefined pattern. For example the pattern in for the first and the third row, to fill with cross. Each game has the same pattern (otherwise, the robot can't learn). For the Pattern Game, the important properties are more related to what was learned in the previous game, rather than the current one.

2) Mirror game: The second game is a Mirror Game. For the Mirror game, the iCub has to play the same move as the previous one by the Human. The Human can play any color he wants, but only on an empty case. For the Mirror game, we had to add one precision for the robot : the *Human and the Robot don't have the same rules*. The Human has to play anywhere one the board, on a free case. The Robot has to reproduce the same move as the Human. This is like a field player and a goal-keeper in football, the game is the same but the rules are not the same for every player. This is a precision that we have to add in the system.

3) Tic-Tac-Toe: The third game is a classic Tic-Tac-Toe (TTT). For the TTT, a legal move is that each player has to play in an empty case, with a different color that the one used by the previous player. For the TTT, the properties involved are mainly the location (free or occupied) and the color and only the current game really matters.

B. Progress of the game

The Robot can learn two different ways. The first one is to play with the Human, where the Human is considered as the *teacher*, and the Robot as the *student*. In the second way, the robot can learn only by watching two Humans playing together, with one of them considered as the *teacher*, and the other one, as the *student*.

The teacher always will have the knowledge of the rules of the game, and will give feedback for every move (the student's move and the teacher's move).

IV. How to learn legal moves ?

This part of the module called *boardGame-Learning* (BGL) has the aim to distinguish legal moves from illegal moves. A legal move is a move authorized by the game, but not necessarily a *"good"* move. For example for a game like Tic-Tac-Toe, a illegal move is to put a different color of the on of the previous player, only on a

 ISBN: 978-1-61208-250-9

TABLE I
DEFINITION OF THE DIFFERENT PROPERTIES, WITH THE NUMBER OF POSSIBILITIES FOR EACH (G : NUMBER OF GAMES PLAYED, C : NUMBER OF TURNS PLAYED DURING THE CURRENT GAME)

Properties	Definition	Pos.
Location	Free / Occupied	2
Location	Which turn this spot has been played in the current game	C
Location	How many time this spot has been played in the previous games	G
Piece	How many time this piece has been played in the previous games	G
Piece Location	How many time this piece has been played at this location in the previous games	C
Color	Same as previous Different from previous	2
Color Piece	How many time this piece has been played with this color in the previous games	G
Color Location	How many time this piece has been played at this location in the previous games	G

unoccupied case.

The learning system is based on a feedback from the user. We start with the idea that the robot doesn't know the game at all. He just knows that he has different kind of pieces (for example : "pawn", "bishop"... In the case of Tic-Tac-Toe, he only has "pawn"), and a board in front of him and he has to move a piece of any color on the board. The robot also has the knowledge of turn taking.

A. Random trials

At the beginning of the game, the iCub will explore the board and his possibilities, and he will expect a feedback from the Human if his move is legal or not. With this feedback, the robot will be able to extract pertinent properties of the legal moves. Each time a move is played, the robot will increment his statistics according to the properties of the move. The properties concerned are the location, the color, the kind of piece and interaction between these properties (see Table I) :

B. Probabilistic Exploration

Each turn, the iCub will pick a move, according to the probabilities extracted from his experience. The probabilities of each properties are given by the following formula :

$$S = \frac{\sigma + N_{hits}}{\sigma * P + N_{tries}} \quad (1)$$

where :

- σ is a learning rate constant
- N_{hits} is the number of good moves with this property
- N_{tries} is the number of tries of moves with this property
- S is the score of each property

We can easily see that $N_{hits} \leq N_{tries}$. Also σ is a learning rate which correspond to : *"after how many tries should I be certain of my experience ?"*. A high σ will delay the influence of experience into the decision. The probabilities correspond to the normalization of the scores in the aim to have a sum of the probabilities of every possibility of a property, equal to 1. For the first moves, the iCub will randomly pick a move, but following his*"intuition"*. This *"intuition"* can be improved by the use of threshold set on the probabilities.

In order to improve his exploration, after an illegal move, the iCub will try the same spot but will change the color then, the piece, and if finally the problem comes from the location and not the piece, he will change his location.

C. Sine qua non conditions

After a few tries, the iCub will be able to extract some *sine qua non* conditions. For this, he has to check if for one property : $N_{hits} = 0$; and : $N_{tries} > \theta$, or simply : "After θ moves with a given property, I have always failed, this should be a *sine qua non* condition" not to make this move. And he will set the probability to 0. This assume that after θ tries, all the configuration with one property fixed, have been seen. The iCub will be able to tell with certainty if, according to the

 ISBN: 978-1-61208-250-9

Human a move is legal or not.

The value of θ will be discussed in the paragraph. V-D.

V. SIMULATION

In order to test the system, we proceed with simulation experiments

A. Games used

As described in paragraph III-A, we used 3 different games for our tests : the Tic-Tac-Toe (TTT), a Mirror Game, and a Pattern Game.

B. Simulation of the teacher moves

For the teacher moves, we coded the rules of each game. We encoded a function to pick a legal move, and a function to attribute a reward to a student's move. In this case, the teacher has no strategy, but it doesn't matter, because, we only focus on the legal moves (see paragraph IV).

C. Results

To test our system, we also simulated the Human part. The simulated Human knows the exact rules of each game, and always picks a legal move for each game. We also made a auditor of legal or illegal move to check. But for each new game, we have to implement the rules of the Human and the auditor.

For 200 simulations, we made the iCub and the simulated Human to play together and the results are summarized in the Fig. 3, Fig. 4 and 5.

1) Result Pattern game: As shown on the Fig. 5, the probability of succes with our module increase rapidly, to reach 100%. However, we can see some differences between different values of the parameter θ. We will discuss about this parameter in paragraph V-D.

2) Result Mirror game: For the mirror game, we can see that the curve of the random pick, seems to stabilize at 0.05, which correspond to $\frac{1}{2}$ for the color, multiplied by $\frac{1}{9}$ for the location, i.e. : $\frac{1}{18} \simeq 0.0556$.

The random curves correspond to a random pick from the *iCub* (18 possibility : 9 case and 2 colors).

3) Result Tic-Tac-Toe: For the Fig. 3 we can see that a random pick we be more successful for the first pick. It is just an effect of the game. Indeed, the Human always started to play. After one move, 8 cases were free, and the iCub had a probability of $\frac{1}{2}$ to take the good color, and of $\frac{8}{9}$ to choose a good spot i.e. : $\frac{4}{9} \simeq 0.444$. But this is only for the first move, because the situation changes in each simulation according to the success or not of the robot. This is why for the first random pick, we have a good probability of success.

D. Influence of the θ parameter.

As we explain earlier, θ is a constant that we can see as a time constant. It corresponds to: "when do I assume that I have seen enough moves to apply a threshold on my predictions ?". As we can see in Fig. 3, for the TTT, as soon as this θ is reached, a rate of succes of 100% is also reached. This mean that after 10 moves, the iCub has seen enough to understand. On contrary, for the Pattern game, the results are better for a higher θ (Fig. 5). After only 10 moves, the iCub *"thinks"* that he has seen enough to be sure of his decision, but the game is too complicated to be understood after so few moves. The whole space of possibles is not yet totally explored.

This parameter can be set according to the complexity of the game.

VI. HUMAN INTERACTION

A. Experiment Description

This experiment involves the iCub and the Human physically interacting, playing a game, using objects on the *Reactable* as pieces (see Fig. 7).
The robot knowledge is initialized through the OPC setup. The database includes :

- 11 locations : 9 brown squares (3x3) from A-1 to C-3 and 2 squares for the objects origin place, forming the game "board".

 ISBN: 978-1-61208-250-9

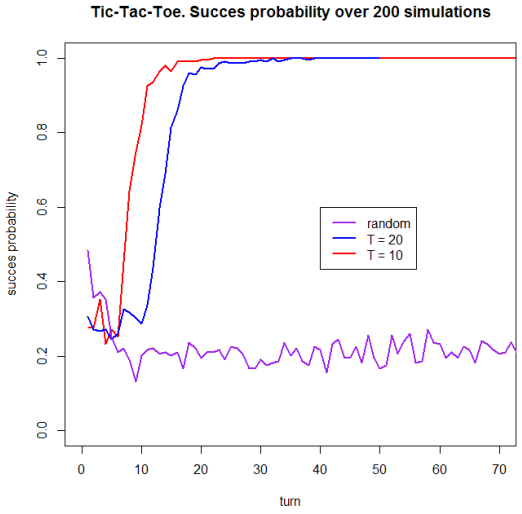

Fig. 3. Results of the learning module on a Tic-Tac-Toe. X-axis is the number of turns passed, Y-axe is the rate of success of the X^{th} move over 70 simulations. The blue curves is for $T = 20$, red for $T = 10$ and the purple curve is in the case of a totally random pick. T is the delay before checking the *sine qua non* conditions.

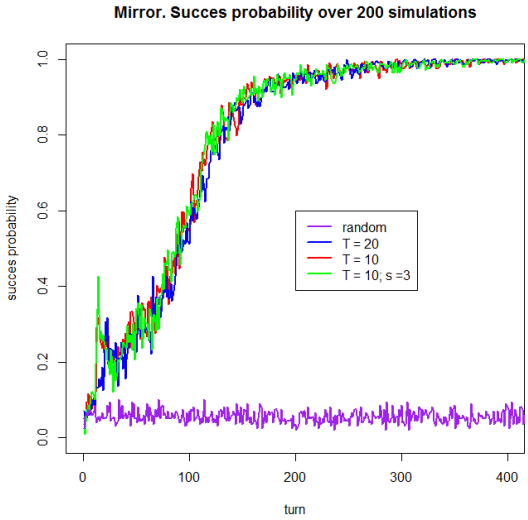

Fig. 4. Results of the learning module on Mirror game. X-axe is the number of turns passed, Y-axis is the rate of success of the X^{th} move over 200 simulations. The blue curves is for $T = 20$, red and green for $T = 10$ and the purple curve is in the case of a totally random pick. T is the delay before checking the *sine qua non* conditions. All the curves are set with $\sigma = 6$ except the green curve for the mirror game where $\sigma = 3$.

- 3 objects : a cross (red) and a circle (blue), which are the played pawn and will change the color of the squares where there are. An eraser is also available to do some correction (i.e. change the color of the square) on the board in case of mistakes.

The objects are like stamps, or playing pieces to be put on the board. To take a turn, the player need to put the object he wants to use from the origin place to the location wanted. An algorithm check where a location is intersected with an object and will change his color accordingly. The user, in case of human, has to say it has moved a pawn ("Done") in order to detect what is the move played by the user (What pawn has been placed where?). It is automatically launched when the iCub is playing at the end of his move. Next the user has to bring back the piece to its corresponding origin location and invite the other player to play when it is done. If the iCub has played, and if it is in a learning mode, the robot asks the human to give a feedback about the move (illegal, bad, good or draw/win at the end of a game). If the move is illegal, the iCub will then play again until he find a proper one.

B. Generality and perspectives

The iCub can also learn just by watching 2 Humans playing, but he will not be able to extract some *sine qua non* conditions, if the Humans always play some legal moves (N_{hits} will always be N_{tries}, except if one of the Humans is a beginner too).

We have seen that we give to the iCub 2 presuppositions : the turn taking, and if the 2 players have the same rules (mirror game) or not. The concept of turn taking, as it is learned in Broz et al.[16] might be interesting to include in the robot. But this is a very primitive instinct and it might be much more interesting for us, to learn the different roles of each player.

 ISBN: 978-1-61208-250-9

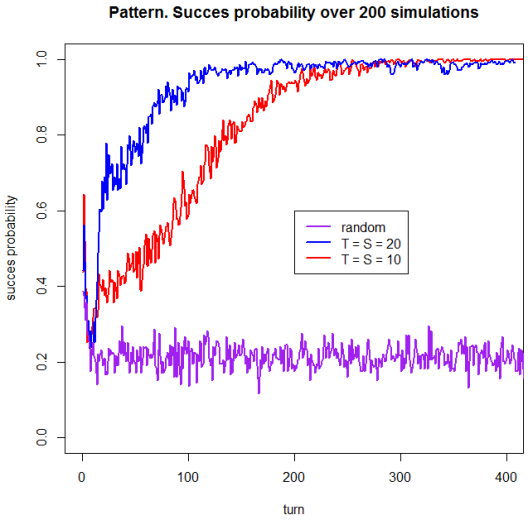

Fig. 5. Results of the learning module on Pattern game. The pattern to find was three crosses on the left and right column, and three circles in the middle column. X-axe is the number of turns passed, Y-axis is the rate of success of the X^{th} move over 200 simulations. The blue curves is for $T = \sigma = 20$, red and green for $T = \sigma = 10$ and the purple curve is in the case of a totally random pick. T is the delay before checking the *sine qua non* conditions.

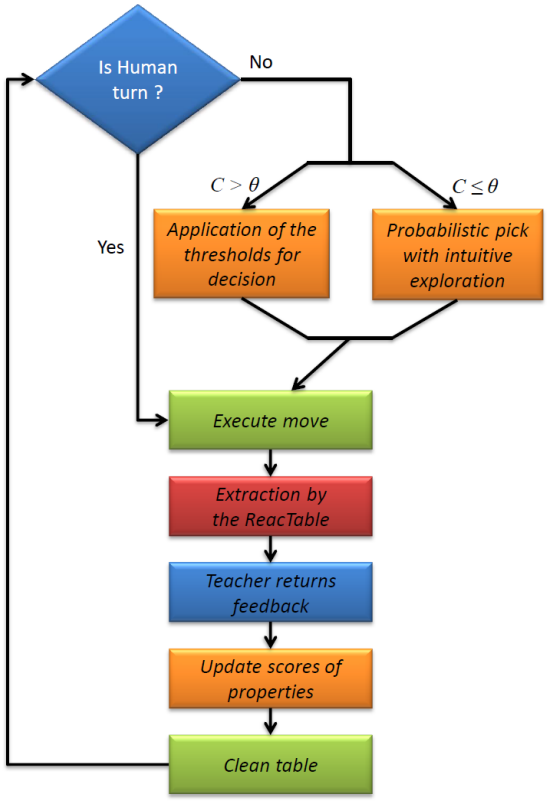

Fig. 6. Progression of moves with our learning system. C is the number of move since the beginning of the learning. θ is the confidence threshold. It can have one or two humans playing. One of them is always the teacher and will give the feedback for each turn (legal-illegal. Win-lose.). The Human/Teacher actions are in blue. In orange is the computation part of the system. In red is the acquisition with the *Reactable* of each move. In green is the player (Robot or Human) action : play and clean. The "execute action" can be done by the three kind of player : Teacher, Human and Robot.

VII. Conclusion and futur work

We have been able to manage the learning of different board game for our robot, simply by using the experience, and intrinsic properties of action. But a problem remains in the learning of "good" or "bad" moves. For this we are currently developing a system based on experience and on the outcome of each game to learn the winning combination and not to be tempted to reproduce a failing behaviour.

In our case, the tasks that the system can be used to solve were games, but we can imagine to learn cooperation as is the case for the Pattern game. The long term aim of this system, is not only to learn board game, but to understand more complex concepts. For example, with the Pattern Game, we can teach some concepts as : *"a row", "a line", "left", "right", "before", "after"...* All concept, put together, can reach to emergence of more complex rules, or understanding of new concepts.

Acknowledgment

This research was supported by the EFAA Project, funded by FP7-ICT- Challenge 2 Cognitive Systems, Interaction, Robotics Grant Agreement no: 270490

 ISBN: 978-1-61208-250-9

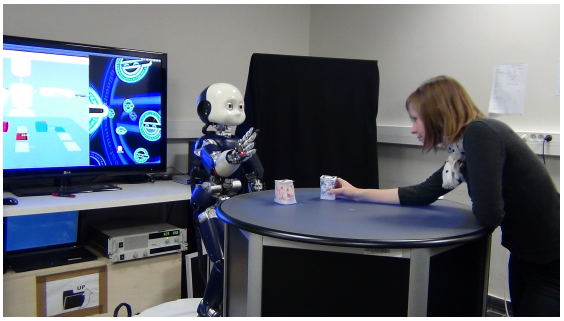

Fig. 7. Normal working of the module with a subject and the Reactable.

REFERENCES

[1] C. Nehaniv and K. Dautenhahn, "Of Hummingbirds and Helicopters: An Algebraic Framework for Interdisciplinary Studies of Imitation and Its Applications." *Interdisciplinary Approaches to Robot Learning*, vol. 24, 2000.

[2] M. Petit, S. Lallee, J. Boucher, G. Pointeau, P. Cheminade, D. Ognibene, E. Chinellato, U. Pattacini, I. Gori, U. Martinez-Hernandez, H. Barron-Gonzalez, M. Inderbitzin, A. Luvizotto, V. Vouloutsi, Y. Demiris, G. Metta, and P. Dominey, "The Coordinating Role of Language in Real-Time Multi-Modal Learning of Cooperative Tasks," *IEEE TAMD*, pp. 3539–3544, 2012.

[3] X. Wu and J. Kofman, "Human-Inspired Robot Task Learning from Human Teaching," *2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA*, 2008.

[4] C. Breazeal and A. Thomaz, "Learning from Human Teachers with Socially Guided Exploration," *2008 IEEE International Conference on Robotics and Automation*, pp. 3539–3544, 2008.

[5] D. Grollman and O. Jenkins, "Dogged Learning for Robots," *2007 IEEE International Conference on Robotics and Automation, Roma, Italy*, 2007.

[6] 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. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 37, 2007.

[7] N. Mirza, C. Nehaviv, K. Dautenhahn, and R. Boekhorst, "Developing Social Action Capabilities in a Humanoid Robot using an Interaction History Architecture," *Proc. IEEE-RAS Humanoids 2008*, 2008.

[8] S. Lalle, U. Pattacini, J. Boucher, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, K. Hamann, J. Steinwender, E. Sisbot, G. Metta, R. Alami, M. Warnier, J. Guitton, F. Warneken, and P. Dominey, "Towards a Platform-Independent Cooperative Human-Robot Interaction System: II. Perception, Execution and Imitation of Goal Directed Actions Proceedings," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[9] G. Geiger, N. Alber, S. Jordà, and M. Alonso, "The Reactable: A Collaborative Musical Instrument for Playing and Understanding Music," *Her&Mus. Heritage & Museography*, no. 4, pp. 36 – 43, 2010.

[10] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, *The iCub humanoid robot: an open platform for research in embodied cognition*. New York, New York, USA: ACM Press, Aug. 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1774674.1774683

[11] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet Another Robot Platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, 2006. [Online]. Available: http://eris.liralab.it/yarp/

[12] V. Mohan, P. Morasso, G. Metta, and G. Sandini, "A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots," *Autonomous Robots*, vol. 27, pp. 291–307, 2009.

[13] M. Kaltenbrunner and R. Bencina, "reacTIVision: a computer-vision framework for table-based tangible interaction," in *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*. New York, New York, USA: ACM Press, Feb. 2007, p. 69.

[14] R. Bencina and M. Kaltenbrunner, "The Design and Evolution of Fiducials for the reacTIVision System," in *3rd international conference on generative systems in the electronic arts*, 2005.

[15] S. Sutton, "Predicting and explaining intentions and behavior: How well are we doing?" *Journal of Applied Social Psychology*, vol. 28, pp. 1317–1338, 1998.

[16] F. Broz, H. Kose-bagci, C. L. Nehaniv, K. Dautenhahn, and A. V. Attention, "Learning behavior for a social interaction game with a childlike humanoid robot."

 ISBN: 978-1-61208-250-9