Learning to interact with humans using goal-directed and habitual behaviors

Erwan Renaudo[1,2], Sandra Devin[3,4], Benoît Girard[1,2], Raja Chatila[1,2], Rachid Alami[3,5], Mehdi Khamassi[1,2], and Aurélie Clodic[3,5]

[1]Sorbonne Universités, UPMC Univ Paris 06, UMR 7222, ISIR, F–75005, Paris, France
[2]CNRS, ISIR, UMR 7222, F–75005, Paris, France
[3]CNRS, LAAS, 7 avenue du colonel Roche, F–31400 Toulouse, France
[4]Univ de Toulouse, INP, LAAS, F–31400 Toulouse, France
[5]Univ de Toulouse, LAAS, F–31400 Toulouse, France

***Abstract*— In order to improve adaptation capabilities of robots for human-robot interaction, we take inspiration from psychology and neuroscience to propose a hybrid control architecture. This architecture is based on the multiple Experts approach that is mainly used for mammal behavior modelling. We propose to couple a human-aware task planner (HATP) with a model-free reinforcement learning to allow the robot to learn behaviors relevant to solve tasks in interaction, taking advantage from the a-priori knowledge provided to the planner and the cheap decision capability of the reinforcement learning agent. We evaluate this architecture in a HRI task of cleaning a table and show that the combination of Experts (planner and reinforcement learning agent) increases the learning speed of the learning agent.**

I. Introduction

Studies on mammals in psychology and neuroscience have provided strong evidence that two categories of behavior exist and mammals switch between them [1], [2]. Goal-directed behaviors – relying on costly but adaptive planned long-term consequences of actions [3] – and habitual behaviors – reactive behaviors fitted to a stable given context – have been extensively studied [4], modeled as Experts using the theory of Reinforcement Learning [5]. The computational principles underlying the switch between behaviors is an ongoing research topic [6], [7], [8] and some applications of these principles to robotics [9], [10] have produced promising results both for modelling the underlying mechanisms and improving robot capabilities.

On the other hand, the emergence of robots as collaborators of human users increases the need for them to be given robust adaptation capabilities. For a more efficient interaction, robots should be able to perform correctly on a collaborative task no matter if the user is known and predictable or new and unreliable. One key point to allow robots to interact with other agents is to integrate them in the planning process. This has been achieved by Human Aware Task Planner (HATP) [11] that, among others, outputs a plan with both the robot's and the other agent's action flows, some actions being joint and some being independent.

In this paper, we apply behavioral architecture principles in a human-robot interaction task (simulated "table cleaning" task [12]), in order to verify both the generality of the multiple behaviors approach and its efficiency for interaction and adaptability to various human users. We propose an architecture with HATP as goal-directed behavior and a model-free reinforcement learning algorithm (MF) as habitual behavior. Whereas the HATP planner embed knowledge on how to solve the task in interaction, its ability to plan bootstraps learning of the model-free algorithm. When the latter has learnt, the robot can rely on its cheap decision capability to avoid planning but still perform correctly. As a proof of concept, we evaluate the performance of each behavior controlling alone the robot, and a random mixture of HATP and MF propositions. We show that HATP performs better than MF on the task, as it starts with an initial knowledge. The combination of HATP and MF improves the performance of the robot on the task, widening the results from [10] on a different, more complex task. The MF actually learns faster, taking advantage from HATP knowledge and guidance.

II. Control architecture

A. Experts

We have two systems able to provide the next action to do to the robot:

- Human-Aware Task Planner (HATP) [11]. This planner allows the robot to take into account a human partner in a cooperative context. As a HTN planner, HATP uses known preconditions and effects of actions in order to find the best plan that reaches the goal. It takes as input a list of all possible actions and their description in terms of preconditions and effects and also a description of the current world state as a set of predicates with a closed world assumption. Then, it looks for the combination of actions that minimizes the solution cost. This cost is computed based on execution time and human-aware costs, e.g the balance of efforts between agents or waiting time of the human partner. This plan is then executed step by step until the goal is reached. It is thus well-suited to be the Goal-Directed Expert in our architecture.
- Qlearning algorithm (MF). It is a one layer neural network implementation of a Qlearning algorithm, a model-free reinforcement learning algorithm [5]. From the activity generated by the current state, each action receives a value depending on the connection weights between input neurons and output (action) neurons. These values are converted into probability of selection using the Boltzmann function. Each action performed is valued by a feedback, strengthening or decreasing the connection between state and action, thus the probability to select each action in a certain state evolves over time, in order to maximize the accumulation of this feedback signal.

Both systems have advantages and drawbacks, but are in fact complementary:

- HATP has a priori knowledge, provided by a human expert, stored in the task domain. It provides a complete plan and it does not require a learning phase. However, it can happen that it is not able to find a plan. Moreover, it becomes slower to provide a solution as soon as the domain gets bigger or the combinatorics increases.
- The Qlearning algorithm requires a learning phase that could be long but it is always able to propose an action. Moreover, once the learning phase has been realized, the system proposes the most interesting action known without planning, and thus is very fast. On the other hand, if the task changes, learning has to be done again according to the new conditions, which is even longer than learning from scratch.

Our idea it to find how to combine them to take advantage of both. In this paper, we will explain how we use HATP planning system to bootstrap the Qlearning algorithm. This has two goals, the first one is obviously to fasten the learning phase of the algorithm, the second one is to enable the learning algorithm to learn a policy that is quite consistent in order to avoid too many inconsistent behavior of the system when it will be faced to a real human.

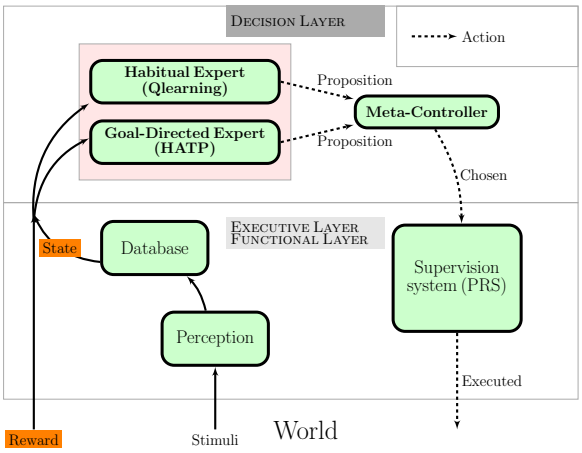

Fig. 1. The control architecture of the robot. The Decision layer is a multiple Experts architecture, arbitrated by the MetaController. In parallel, the Goal-directed Expert, which is long to plan but can take action consequences into account, and the Habitual Expert, which is quick to plan but slow to learn the policy, propose an action. The final decision (chosen action) is sent to the Executive and Functional Layers for execution.

B. Architecture

We adapt the model from [10] to the layered architecture from [13] except that in the Decision Layer, the Value Iteration Model-based algorithm is replaced by the Human-Aware Task Planner (HATP). Consequently we have two Experts: one called Goal-Directed Expert that is hold by HATP, one called Habitual Expert that is hold by Qlearning. Figure 1 gives an overview of the architecture.

Experts receive the current state estimation of the world (it matches the state description, see section III) from the perception processing, they both receive exactly the same information. They then send their next action proposition to the MetaController. The MetaController arbitrates between the two propositions and send the chosen action to the supervision system to be executed. The Habitual Expert receives back the executed action in order to learn. The system loops at the end of the action execution.

In this setup, we only use a random arbitration between Experts propositions. Few of the criteria proposed

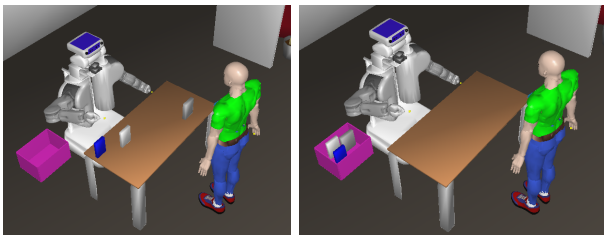

Fig. 2. The Robot and the human partner are facing the table. One tape is on human side, two tapes and the trashbin are on robot side. At left, the initial state. At right, the final state.

in [14] are directly applicables, as they mostly rely on the probability distributions maintained by their Experts for decision. Using HATP implies that the MetaController has no probability information on the action proposed by the goal-directed Expert. Thus, exploration of relevant criteria will be discussed in section V.

III. Task Description

The task is inspired from [12]. The goal of the task is for the robot to *clean the table* with the help of a human partner. Cleaning the table is equivalent to "all items are in the thrashbin" (or none are on the table - but the first expression of the goal is more restrictive on the final state to be in). Some of the objects can be out of its range, so the human partner should help. Figure 2 shows the setup.

The task has been run in simulation for now, but the running algorithms to pilot the robot are the same than the ones running on the real robot.

We construct the state manipulated by the architecture as a vector of the following boolean facts on environment. These facts are computed based on geometry in [15].

- Object isReachableBy Robot (including Trashbin)
- Object isReachableBy Human (including Trashbin)
- Object isIn Trashbin
- Human isReachableBy Robot
- Robot hasInHand Object
- Human hasInHand Object

These facts are computed from the point of view of the robot.

The set of actions that are available for both actors are the following: PickObject, ThrowObject, GiveToHuman = (give(from robot), take(from human)), TakeFromHuman =(give(from human), take(from robot)), Wait, Goto (Object, Trashbin, Human).

The Wait action allows the robot to wait until the human performs an action (or until a time limit is reached). HATP returns this action when, according to its plan, the next robot action cannot be executed without human intervention (e.g. only one tape remains, out of robot range). In simulation, we give to the human a "collaborative behaviour": he performs the actions that HATP planned for him and participates to a handover when the robot needs to. Thanks to this action, the robot will indirectly learn the behaviour of the human partner. If the behaviour of the human changes (e.g. he becomes less cooperative or has a different way to solve the task), the wait action will not have the same effects and so, the robot will learn to achieve the task in a different way.

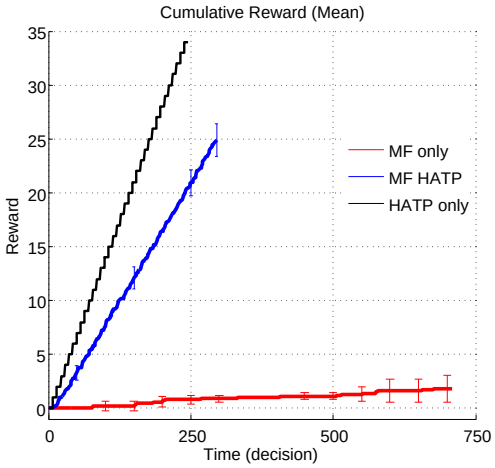

Fig. 3. Mean cumulative reward on 10 simulations where the robot repeatedly fulfils the task. Errorbars represent the standard deviation from the mean every 100 decisions.

IV. Results

We evaluate the performance of the robot using the cumulative reward obtained during the task duration. One reward unit is given to the robot when the table is empty and the robot waits. The setup is then reinitialized and the task can be done again, thus the cumulative reward corresponds to the number of times the task has been solved. We study the performance of each Expert (HATP and MF) controlling the robot alone, and then the combination of them. All experiments last the same fixed time, but the number of decisions taken at the end may vary.

From figure 3, we observe the performance of each Expert alone and the combination by looking the number of rewards during a fixed duration (i.e. the number of times the system is able to solve the task).. We observe a poor performance of the MF alone, which is not able to solve the task more than three times. As the MF has no initial knowledge, it has to discover the right sequence

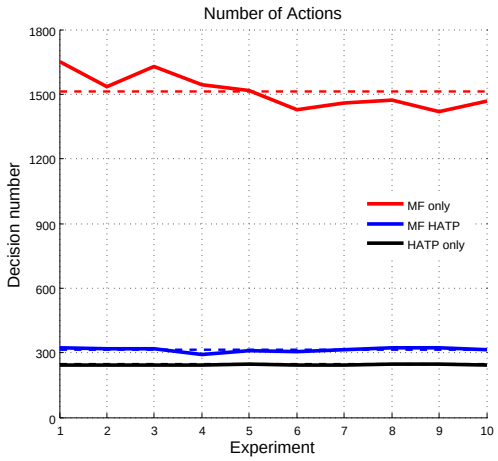

Fig. 4. Number of action per experiment. Dashed line is the mean number of action depending on the control method (MF only, HATP only or combination).

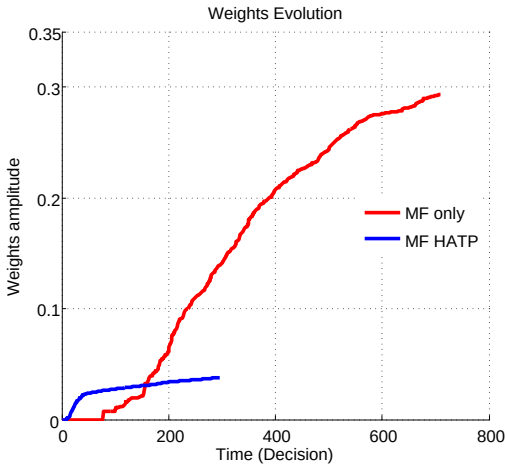

Fig. 5. Mean MF connection weights evolution for MF alone and MF and HATP combination. The amplitude is defined as the sum of the absolute value of weights. Weights are initialized to zero, thus the higher the amplitude is, the more the MF has learnt which action to do.

of actions, which is non trivial as the number of states and actions is quite big.

The random combination HATP-MF is performing much better than the MF alone, solving the task 25 times in average. However, HATP alone perform even better, solving the task 34 times in average. Indeed, the task is easy enough to solve for HATP and the time required to find a plan is negligible here. As the simulated human always performs the actions planned by HATP, the plan found by HATP is always optimal and will never change during the task execution. Accordingly, the random combination of HATP and the MF performs worst as it can include actions proposed by the MF that make the plan non optimal.

Figure 4 shows the number of actions proposed to the supervision system during each experiment. We can see that the MF alone suggests twice to three times more actions than HATP or the combination in the same given time. This is mainly due to the way each Expert decides: the MF only needs to compute the values of each action (which is propagating the state activity to action neurons) and to draw an action from the resulting probability distribution. It proposes a lot of infeasible actions and the supervision system will not spend time to execute them as it will stop to the preconditions verifications. HATP checks for action preconditions when planning and so, for each of the action proposed by HATP, the supervisor spend time to execute it (or try to execute it if the action is not really feasible according to the geometry). The number of actions suggested by the combination of Experts is closer to the one with HATP alone while remaining lightly higher. It can be explained by the fact that a part of the actions proposed by the combination comes from HATP and, for the ones coming from the MF, HATP helps it to learn faster a solution, causing it to propose less infeasible actions.

Finally, we analyse the effect of combination on learning of MF in Figure 5. Learning is evaluated by weights amplitude, namely the sum of weights absolute value over actions. The MF starts with weights initialized to zero, each learning step increases or decreases the value of some of the weights, until convergence. The figure shows that learning occurs much earlier for the combination of Experts than when the MF is alone. The combination has a bootstrapping effect and the knowledge about the task from HATP is transfered to the MF. This shows that a human-provided a priori knowledge can be used to guide exploration and learn quicker. Even if not tested in this experiment, this means that a change in task condition for which HATP can find a new plan can be learnt quickly by the MF, so the robot will be able to adapt to the new conditions without taking too much time.

V. Discussion

We proposed the Decision Layer of a robotic control architecture combining two Experts: a human-aware planner (HATP) and a model-free reinforcement learning agent (MF). We evaluated each Expert on a simulated ”table cleaning” task in interaction with a human. We showed that combining the Experts’ action proposition significantly increases the performances and the learning speed compared to the MF alone. In term

of pure performance, the combination performs worst than HATP alone in this task. However, we strongly believe that having a combination of Experts compared to HATP alone will be beneficial even if the performance decreases. Indeed, the addition of the learning agent will allow the system to find a solution in some specific cases where the planner can be blocked (e.g. HATP does not take into account the geometry and can find plan feasible at a symbolic level but not in reality). Moreover, the learning agent will allow the robot to adapt its behaviour if the human does not act as expected by HATP. In future work we will test the system with a more complex task that will challenge more HATP and with different behaviours for the human.

Here, we used a random criterion as a proof of concept: it is a starting point to validate our principle on a much more realistic task than in [10]. However, in order for a human being to trust the robot enough to collaborate on a task, the behavior can not be random when the MF is still learning. Thus the need for a relevant criterion implies that the latter is able to take into account the state of learning of the MF and gives it control only when learning has converged into a stable policy. Give to the MetaController an information on the variation of MF weights can allow for a more clever arbitration.

In our architecture, we start with a difference between the initial knowledge provided to each Expert. We could provide an equivalent knowledge to the MF, which would be able to find a good behavior since the beginning. However, we choose not to do it because this knowledge, as is the knowledge given to HATP, would be human-provided, and thus will not be necessarily well-adapted to the robot. For example, the knowledge in HATP may consider that the task can be solved by taking the tape out of robot's range from the human partner giving it. It makes the assumption that any human partner will actively participate in solving the task. But the one that provided this knowledge neglects that the partner may not understand what is expected from her or may be afraid of the robot. Providing the same – potentially wrong – knowledge to the MF will prevent it to try unexpected behavior, explore the task space and eventually find a better-suited solution for a real world problem.

HATP is indeed limited to the knowledge the human gave to it. If it happens that a case has been forget by its designer, HATP is blocked. The use of simulation could help us to find these kind of limitations and the use of the learning algorithm could help to find a suited solution that then could be added to HATP domain. HATP is able to devise a plan not only for the robot but also for the human it interacts with. It is well suited for interaction because it can take into account general knowledge about how the task should unfold from a human perspective (and predictability is something you are looking at when you interact with somebody). However, if we think about interaction, we have also to think about adjustment given the human the robot interacts with. Flexibility of learning could help in that direction. It could help, for example, to tune costs used by HATP to compute its plan so it is better suited for interaction with a given person.

In order to validate more generally this approach, the experiment will be done on a real robot and with real human partners interacting, as our architecture is easily transferable from simulation to real world. This will provide some behavioral variability from non-simulated human partners. We will also be able to study different ways to give reward to the robot: it is given from the simulation in this work, but it could be extracted from analysing the reaction from robot's partner and increase robot's autonomy.

ACKNOWLEDGEMENTS

This work has been funded by a DGA (French National Defence Agency) scholarship (ER), by the Project HABOT from Ville de Paris and by French Agence Nationale de la Recherche ROBOERGOSUM project under reference ANR-12-CORD-0030.

REFERENCES

[1] A. Dickinson, "Actions and habits: The development of behavioral autonomy," *Philosophical Transactions of the Royal Society (London)*, vol. 308, pp. 67 – 78, 1985 1985.

[2] B. W. Balleine and J. P. O'Doherty, "Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action.," *Neuropsychopharmacology*, vol. 35, pp. 48–69, 2010.

[3] B. W. Balleine and A. Dickinson, "Goal-directed instrumental action: contingency and incentive learning and their cortical substrates.," *Neuropharmacology*, vol. 37, pp. 407–419, 1998.

[4] R. J. Dolan and P. Dayan, "Goals and habits in the brain," *Neuron*, vol. 80, no. 2, pp. 312 – 325, 2013.

[5] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[6] N. Daw, Y. Niv, and P. Dayan, "Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control.," *Nature Neuroscience*, vol. 8, no. 12, pp. 1704–1711, 2005.

[7] M. Keramati, A. Dezfouli, and P. Piray, "Speed/accuracy trade-off between the habitual and goal-directed processes.," *PLoS Computational Biology*, vol. 7, no. 5, pp. 1–25, 2011.

[8] A. Dezfouli and B. W. Balleine, "Habits, action sequences and reinforcement learning," *European Journal of Neuroscience*, vol. 35, no. 7, pp. 1036–1051, 2012.

[9] K. Caluwaerts, M. Staffa, S. N'Guyen, C. Grand, L. Dollé, A. Favre-Félix, B. Girard, and M. Khamassi, "A biologically inspired meta-control navigation system for the psikharpax rat robot," *Bioinspiration & Biomimetics*, 2012.

[10] E. Renaudo, B. Girard, R. Chatila, and M. Khamassi, "Design of a control architecture for habit learning in robots," in *Biomimetic and Biohybrid Systems - Third International Conference, Living Machines 2014, Milan, Italy, July 30 - August 1, 2014. Proceedings*, pp. 249–260, 2014.

[11] R. Lallement, L. de Silva, and R. Alami, "Hatp: An htn planner for robotics," in *2nd ICAPS Workshop on Planning and Robotics*, pp. 20–27, 2014.

[12] R. Alami, M. Warnier, J. Guitton, S. Lemaignan, and E. A. Sisbot, "When the robot considers the human...," in *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[13] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *I. J. Robotic Res.*, vol. 17, no. 4, pp. 315–337, 1998.

[14] E. Renaudo, B. Girard, R. Chatila, and M. Khamassi, "Which criteria for autonomously shifting between goal-directed and habitual behaviors in robots?," in *the Fifth Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, (Japan), p. to appear, Aug. 2015.

[15] E. A. Sisbot, R. Ros, and R. Alami, "Situation assessment for human-robot interactive object manipulation," in *RO-MAN, 2011 IEEE*, pp. 15–20, IEEE, 2011.