An adaptive human-robot cooperation framework for assembly-like tasks

Kourosh Darvish, Barbara Bruno
Enrico Simetti, Fulvio Mastrogiovanni, and Giuseppe Casalino

University of Genoa
kourosh.darvish@edu.unige.it
{barbara.bruno,enrico.simetti,fulvio.mastrogiovanni,giuseppe.casalino}@unige.it

Abstract. In this paper we introduce a method for human-robot cooperation, specifically for assembly-like tasks. Novel approaches for cooperation are needed if we want to enable *intuitive* and *natural* interaction between humans and robots in future factories. Our contribution is two-fold: i) a framework for the representation of the cooperation task, which allows for run-time adaptation; ii) a dynamic procedure to monitor task execution based on AND/OR graphs. The framework has been experimentally validated in a cooperation scenario in which a Baxter robot and a human perform a screwing task together.

Keywords: Human-robot cooperation, Future factory, Task representation, Human action recognition, AND/OR graph.

1 Introduction

Human-robot cooperation (HRC) is expected to enhance and broaden the role of robots in many real-world scenarios. A possible example is manufacturing, in which humans and robots can play different roles in performing certain tasks, depending on their capabilities. Our objective is to explore the use of HRC for assembly-like tasks.

Given the high demand for adaptability and flexibility in production lines, one possibility is to avoid any *a priori* defined task structure (i.e., the sequence of operations), and to allow humans to decide which operation to perform on the fly [1]. First examples can be found in recent literature, such as a *static* architecture based on AND/OR graphs [2], a hidden semi-Markov model for task teaching [3], a fixed task-duration AND/OR graph [4].

The robot co-worker must: i) recognise human actions, and specifically their gestures, using a sort of *gesture sensor*; ii) represent all the allowed sequences of operations, and their mutual relationships, which we model using AND/OR graphs [5]; iii) reason about a specific sequence of operations, as performed by the human, i.e., understanding the *cooperation context*; iv) according to the context, select the *most appropriate* action to carry out at run time.

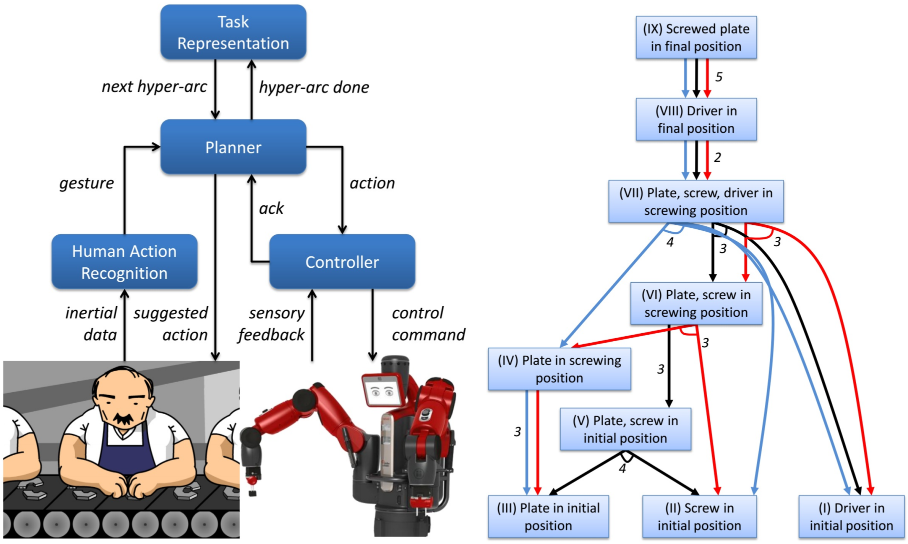

Fig. 1: Left: the system's architecture. Right: an example AND/OR graph.

We propose a framework for HRC (in particular for assembly) tasks allowing for task representation and the run-time adaptation to human actions, by means of the execution of proper robot actions. Such adaptation capabilities are enabled by the robot's ability to assess the cooperation context, the AND/OR graph based task representation formalism and the employed graph traversal procedure to adapt to human actions.

2 System's Architecture and Task Representation

The system's architecture, shown in Figure 1 on the left hand side, includes a *Human Action Recognition* (HAR) module, a *Controller* module, a *Planner* module and a *Task Representation* (TR) module.

TR encodes the cooperation task structure as an AND/OR graph. Furthermore, it interacts with Planner to determine which actions have been performed and to provide alternatives for next actions to carry out for the human or the robot, respectively. HAR is a sort of *gesture sensor* to recognise human actions, introduced in [6]. It relies on inertial data measured at the human's right wrist by means of wearable devices. HAR requires the offline learning of human gestures to detect. Gestures models in HAR are obtained using Gaussian Mixture Modelling and Regression, as *expected* acceleration patterns. Relevant gestures for our scenario include *pick up*, *put down*, and *screwing*. Online, HAR processes the current data stream to provide a unique gesture label. Planner updates the representation with the recognised gesture (retrieved from HAR) and the last robot's actions (retrieved from Controller). Then, Planner provides suggestions for humans or commands for the robot, accordingly.

The Controller is based on a task priority based architecture [7] and reactively handles control tasks of equality (e.g. end-effector position control) and inequality (e.g. joint limits, arm dexterity) type in real time, guaranteeing the robot operability as well as its safety. The Controller accepts commands such as *handover*, *grasp object*, *leave object*, *position object*, as well as *initialise robot* and *stop robot*. Thanks to this approach, the Planner is freed from the burden of many lower-level operative details, making it more efficient and effective.

As we anticipated, TR uses an AND/OR graph to encode the cooperation process. An AND/OR graph $G(N, H)$ is a structure made up of a set N of n nodes and a set H of h hyper-arcs. Nodes in N define reachable *states* in the graph. Hyper-arcs in H define *transition* relationships among nodes. In particular, hyper-arcs define *many-to-one* transition relationships between many *child* nodes and one *parent* node. The relationship between the transitions of a hyper-arc is considered to be in logical *and*, while the relationship between different hyper-arcs of a parent node is in logical *or*. Both nodes and hyper-arcs are associated with costs.

In our case, each hyper-arc h_i models a set of actions A_i, where an action $a_j \in A_i$ can be performed either by a human or a robot during the cooperation process. If the order in which to execute actions in A_i is important, we treat them as a strict *sequence*. Initially, all the actions in A_i are labelled as *unfinished*; when they are executed we label them as *finished*. If all the actions in A_i are finished, then h_i is *done*. Nodes can be either *solved* or *unsolved*. A node $n_k \in N$ is solved if there is at least one hyper-arc h_i to this node, h_i is done and all its child nodes are *solved*. The *leaves* in G are initialised as solved or unsolved at the beginning, depending on the initial state of the cooperation. This procedure iterates going upward to the *root* node of G. When the root is *solved*, then G is labelled as solved. During graph traversal, n_k is *feasible* if there is at least one h_i to it such that all its child nodes are solved. Otherwise, n_k is *unfeasible* and h_i is labelled as *active*. At all times, we have a set of active hyper-arcs $H_a \subset H$ in G.

The *temporal task representation state* S is the set of all the labelled nodes and hyper-arcs in G. S defines the possible action alternatives for the human or the robot in cooperation. We define as *cooperation context* a sequence of actions performed by humans or robots during the cooperation, corresponding to an allowed traversal path in G. When a new human action a_n is detected by HAR, Planner interacts with TR to determine whether a_n belongs to an A_i such that the latter corresponds to an hyper-arc in H_a. If this does not happen, i.e., a_n does not belong to the current cooperation context, the robot enters a *null* mode and waits for further knowledge to determine which traversal path in G is involved. If there are multiple active hyper-arcs possibly involving a_n, the robot enters an *ambiguous* mode. Otherwise, the next action to suggest or perform is defined in G so that the overall cost of the traversal path is minimised.

3 Examples

Figure 1 on the right hand side shows an example AND/OR graph for a cooperative screwing task. The human performs actual screwing, whereas the robot picks up the wooden piece into which screwing occurs and puts it down at the end. There are three possible cooperation contexts. In the Figure, the corresponding paths have different colours (red, blue, and black), and are characterised by different total costs, i.e., 16, 14, and 17 (node costs are zero and hyper-arc costs are expressed in Arabic numbers).

Figure 2 illustrates the first cooperation context. When the robot positions the plate in front of the human (a)-(d), the latter picks up the screw and the screwdriver (e), performs screwing (f), puts down the screwdriver on the table (g)-(h), and finally waits for the robot to finish the assembly task (i)-(l).

Figure 3 depicts the second cooperation context. With the plate located on the table (a), the human puts the screw inside the plate hole (b)-(c), later the robot positions the plate in front of the human (d)-(e), the human picks up the screwdriver (f), performs the screwing (g), puts down the screwdriver on the table (h)-(i), and waits for the robot to finish the assembly task (j)-(l).

It is noteworthy that the framework allows a human to switch between any possible cooperation contexts on the fly simply by performing an action belonging to another context. HAR, TR and Planner continuously monitor human action execution and re-plan a suitable path in G, which leads to a new reference cooperation context.

4 Conclusions

We propose a framework allowing for the representation of cooperative tasks using AND/OR graphs, and their execution leaving a human free to choose the specific sequence of operations to perform among a number of alternatives. The developed architecture is able to adapt to human actions at run-time, which are detected as gestures using wearable devices worn at the human's right wrist. The framework has been validated by modelling a cooperative assembly task, in which a human and a Baxter dual-arm manipulator perform turn-taking actions in any allowed sequence.

References

1. Faber, M., Bützler, J., Schlick, C.M.: Human-robot cooperation in future production systems: analysis of requirements for designing an ergonomic work system. Procedia Manufacturing **3** (2015) 510 – 517
2. Johannsmeier, L., Haddadin, S.: A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. IEEE Robotics and Automation Letters **2**(1) (2017) 41–48
3. Rozo, L., Silvério, J., Calinon, S., Caldwell, D.G.: Exploiting interaction dynamics for learning collaborative robot behaviours. In: Proceedings of the 2016 Interactive

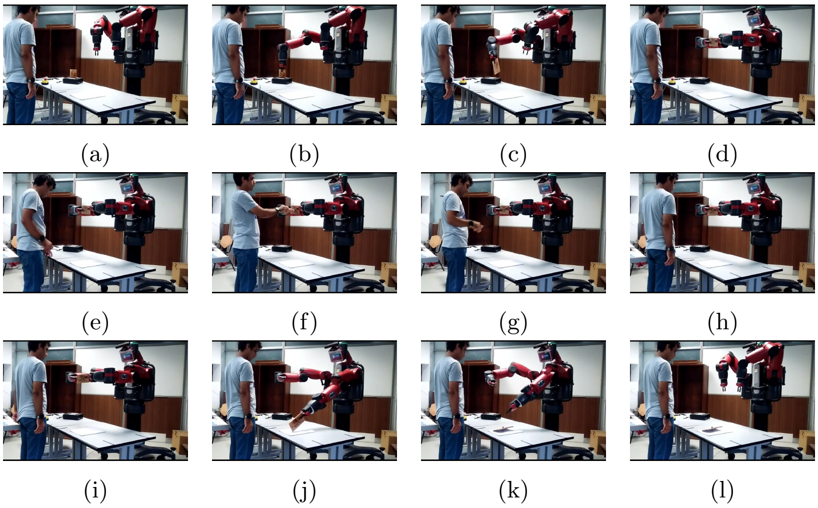

Fig. 2: The cooperation context associated with the red path in Figure 1.

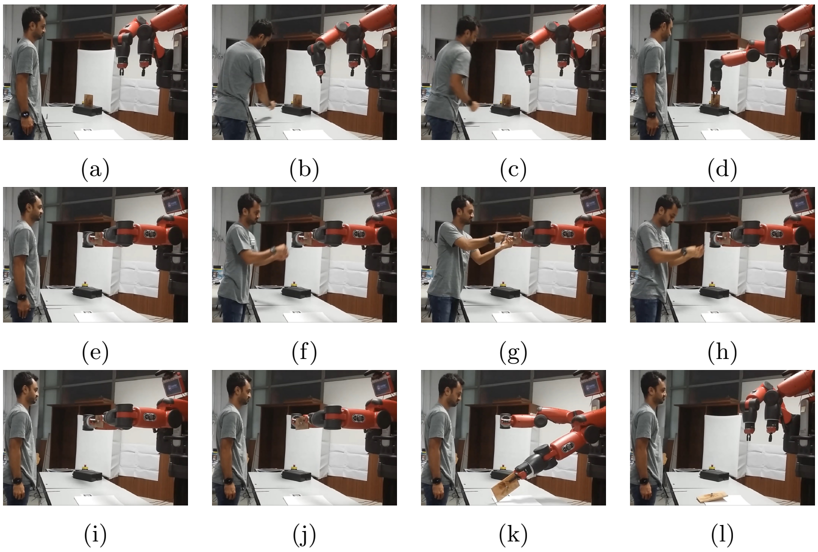

Fig. 3: The cooperation context associated with the blue path in Figure 1.

Machine Learning Workshop, co-located with the 2016 International Joint Conference on Artificial Intelligence (IJCAI 2016), NYC, USA (July 2016)

4. Hawkins, K.P., Bansal, S., Vo, N.N., Bobick, A.F.: Anticipating human actions for collaboration in the presence of task and sensor uncertainty. In: Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014), Hong Kong, China (May 2014)

5. de Mello, L.S.H., Sanderson, A.C.: AND/OR graph representation of assembly plans. IEEE Transactions on Robotics and Automation **6**(2) (1990) 188–199
6. Bruno, B., Mastrogiovanni, F., Sgorbissa, A., Vernazza, T., Zaccaria, R.: Analysis of human behaviour recognition algorithms based on acceleration data. In: Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany (May 2013)
7. Simetti, E., Casalino, G.: A novel practical technique to integrate inequality control objectives and task transitions in priority based control. Journal of Intelligent & Robotic Systems (2016) 1–26