

An Attentional Approach to Human–Robot Interactive Manipulation

Xavier Broquère · Alberto Finzi · Jim Mainprice ·
Silvia Rossi · Daniel Sidobre · Mariacarla Staffa

Accepted: 29 March 2014 / Published online: 18 April 2014
© Springer Science+Business Media Dordrecht 2014

Abstract Human robot collaborative work requires interactive manipulation and object handover. During the execution of such tasks, the robot should monitor manipulation cues to assess the human intentions and quickly determine the appropriate execution strategies. In this paper, we present a control architecture that combines a supervisory attentional system with a human aware manipulation planner to support effective and safe collaborative manipulation. After detailing the approach, we present experimental results describing the system at work with different manipulation tasks (give, receive, pick, and place).

X. Broquère · D. Sidobre (✉)
CNRS, LAAS, 7 avenue du colonel Roche, 31400 Toulouse, France
e-mail: daniel.sidobre@laas.fr

X. Broquère
e-mail: xavier@broquere.fr

X. Broquère
Univ de Toulouse, LAAS, 31400 Toulouse, France

D. Sidobre
Univ de Toulouse, UPS, LAAS, 31400 Toulouse, France

A. Finzi · S. Rossi · M. Staffa
Dipartimento di Ingegneria Elettrica e Tecnologie
dell'Informazione (DIETI), Università degli Studi Napoli
Federico II, via Claudio 21, 80125 Napoli, Italy
e-mail: alberto.finzi@unina.it

S. Rossi
e-mail: silvia.rossi@unina.it

M. Staffa
e-mail: mariacarla.staffa@unina.it

J. Mainprice
Worcester Polytechnic Institute, 100 Institute Road,
Worcester, MA, USA
e-mail: jmainpri@wpi.edu

Keywords Human–robot interaction · Cognitive control · Robot manipulation · Attentional system · Human aware planning and execution

1 Introduction

In order to work with humans, a robotic system should be able to understand the users' behavior and to safely interact with them within a shared workspace. Moreover, in order to be socially acceptable, the behavior of the robotic system has to be safe, comfortable, and natural. In Social Robotics (SR) and Human–Robot Interaction (HRI), object exchange represents a basic and challenging capability [16,20]. Indeed, simple tasks of object handover pose the problem of a close and continuous coordination between humans and robots, which should interpret and adapt their reciprocal movements in a natural and safe manner. From the robot perspective, the human motions and the external environment should be continuously monitored and interpreted searching for interaction opportunities while avoiding unsafe situations. For this purpose, the robotic system should assess the environment to determine whether humans are reachable, attentive, and willing to participate to the handover task. On the other side of the interaction, if the robot movements and intentions are natural and readable, it is easier for the human operator to cooperate with the robot; in this way, the robotic manipulation task can also be simplified by human assistance [20].

During interactive manipulation, sensorimotor coordination processes should be continuously regulated with respect to the mutual human–robotic behavior, hence attentional mechanisms [27,33,35] can play a crucial role. Indeed, they can direct sensors towards the most salient sources of information, filter the sensory data, and provide implicit coordination mechanisms to orchestrate and prioritize concur-

rent/cooperative activities. In this perspective, an attentional system should be exploited not only to monitor the interactive behavior, but also to guide and focus the overall executive control during the interaction.

Attentional mechanisms in HRI have been proposed mainly focusing on visual and joint attention [7,8,28,29,32,39,47]. In these works, the authors introduce and analyze joint visual attentional mechanisms (eye gaze, head/body orientation, pointing gestures, etc.) as implicit nonverbal communication instruments used to improve the quality of the human–robot communication and social interaction. In contrast, we focus our interest on executive attention [36] proposing the deployment of a supervisory attentional system [17,33] that supports safe and natural human–robot interaction and effective task execution during human-aware manipulation. The achievement of this goal is very desirable in SR, where social acceptability and safety earn a role of primary importance.

Our attentional system is designed as an extension of a reactive behavior-based architecture (BBA) [4,9] endowed with bottom-up attentional mechanisms capable of monitoring multiple concurrent processes [27,40]. For this purpose, we assume a frequency-based approach to attention allocation [40] extended to the executive attention. This approach is inspired by [34], where the attentional load due to the accomplishment of a particular task is defined as the quantity of attentional time units devoted to that particular task, and by [40], where attentional allocation mechanisms are related to the sampling rate needed to monitor multiple parallel processes. More specifically, we introduce attentional allocation mechanisms [15], which allow the robot to regulate the resolution at which multiple concurrent processes are monitored and controlled. This is obtained by modulating the frequency of sensory sampling rates and the speed associated with the robot movements [14,15,24]. Following this approach, we consider interactive manipulation tasks like pick and give, receive and place, or give and receive. In this context, the attentional allocation mechanisms are regulated with respect to the humans' dispositions and activities in the environment, taking into account safety and effective task execution. The human–robot interaction state is monitored and assessed through costmaps [30], which evaluate HRI requirements like human safety, reachability, interaction comfort, and field of view. This costmap-based representation provides a uniform assessment of the human–robot interactive state, which is shared by the motion planner and the attentional executive system. Indeed, the costmap-based representation allows the robot manipulation planner and arm controller to generate and to execute human-aware movements. On the other hand, the attentional executive system exploits the cost assessment to regulate the strategies for activity monitoring, action selection, and velocity modulation.

In this paper, we detail our approach presenting a case study along with preliminary empirical results used to show how the system works in typical scenarios of object handovers.

2 Attentional and Safe Interactive Manipulation Framework

In this work, we present an attentional executive system suitable for safe and effective human–robot interaction during cooperative manipulation tasks. We mainly focus on handover tasks and simple manipulation behaviors like pick, place, give, and receive. Here the attentional system is used to distribute the attentional focus on multiple tasks, humans and objects (i.e., the relevant action to perform and the human/object to interact with), to orchestrate parallel behaviors, to decide on task switching, and to modulate the robot execution.

Our approach combines the following design principles:

- *Attentional Executive System*: we deploy attention allocation mechanisms for activity monitoring, action selection, and execution regulation;
- *Spatial and cost-based representation of the interaction*: a set of costmaps functions is computed from the human kinematics state to assess human–robot interaction constraints (distance, visibility, and reachability);
- *Adaptive human-aware planning*: adaptive and reactive human-aware motion/path/grasp planning and replanning techniques are used to generate and to adjust manipulation trajectories. These can be adapted at the execution time by taking into account the costmaps and the attentional state.

Figure 1 details the corresponding attentional framework. The spatial reasoning system allows the robot to assess

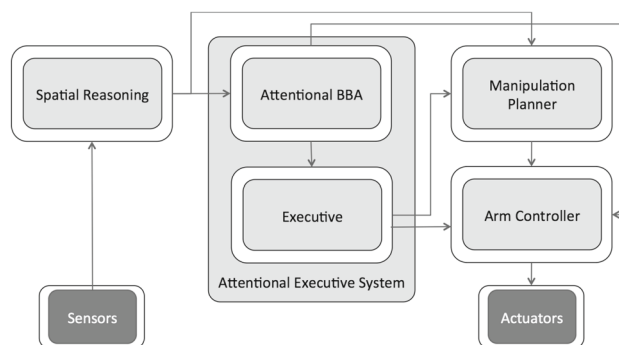


Fig. 1 The spatial reasoning module updates the costmaps used to assess the human posture and behavior. Given the costmap values, the attentional system continuously modulates the behaviors sampling rates and activations. The attentional state is then interpreted by an executive system, which decides about task switches and modulates execution velocity affecting the manipulation planner and the arm controller

human–robot interaction constraints providing interaction costmaps. These costmaps are then used by the attentional executive system and by the human-aware planner to generate safe and comfortable robot trajectories. More precisely, given the costmaps assessment from the human posture and behavior, the attentional behavior-based architecture (attentional BBA) continuously modulates the sensors sampling rate and the actions activations; while, depending on suitable attentional thresholds, the executive system selects the current task inducing path/motion replanning. When the task changes, the executive system aborts the current motion and starts the replanning process. Finally, the arm controller is to execute the trajectory generated by the manipulation planner modulating the velocity as suggested by the attentional executive module. In the following, we detail each component of the architecture.

2.1 Spatial Reasoning

The attentional supervisory system is provided with a rich data set by the spatial reasoning system such as distance, visibility, and reachability assessment for the humans in the scene. This key reasoning capacity enables to perform situation assessment for interactive object manipulation [45] and to determine whether humans are reachable, attentive, and willing to participate to the handover task.

The spatial reasoning module also evaluates the robot interaction space and opportunities in the same manner. This enables to assess the possible manipulation tasks that the robot can achieve alone.

Each property is represented by a human or robot centric costmap that establishes if regions of the workspace are distant, visible or reachable by the agent. All costmaps are computed off-line as arrays of values named grid in the following. They are constructed by considering simple geometrical features such as the distance between a segment and a point or the angle between two vectors (details further). When assessing the cost of a particular point, the value is not computed on the fly but simply looked-up in the preloaded grid. Hence, the attentional system is able to quickly determine whether objects are visible by the human or not by simply reading the value in the costmap. Other examples might be to determine whether an object is reachable or not by a human, whether a human is attentive during handover tasks by considering robot center visibility or whether he/she is too close for handing an object (i.e. the human current position cannot yield a safe handover).

The distance costmap, depicted in Fig. 2a, is computed using a function $f(h) \rightarrow (p_1, p_2)$, which returns two points of interest (p_1 at the head and p_2 at the feet) given a human model h . The two points p_1 and p_2 are then used to define a simplified model of the human composed of a segment and

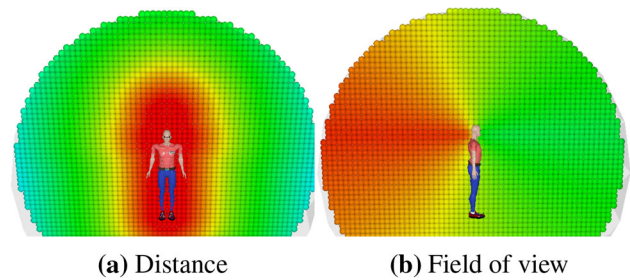


Fig. 2 The human-centered distance costmap (a) and the field of view costmap (b)

a sphere of radius $R = 0.3m$. The distance cost $c_{dist}(h, p)$ between a point p and this simplified model will be:

$$c_{dist}(h, p) = \min(d_s(h, p), \max(0, \|p_1 - p\| - R)) \quad (1)$$

with:

$$d_s(h, p) = \begin{cases} \frac{(p - p_1) \wedge (p_2 - p_1)}{\|p_2 - p_1\|} & \text{if } 0 < \rho < \|p_2 - p_1\| \\ \|p_1 - p\| & \text{if } \rho \leq 0 \\ \|p_2 - p\| & \text{if } \rho \geq \|p_2 - p_1\| \end{cases} \quad (2)$$

where $\rho = (p_1 - p) \cdot \frac{p_2 - p_1}{\|p_2 - p_1\|}$.

This costmap models a safety property as it contains higher costs for regions that are close to the humans. This property is accounted at several levels of the robot architecture to ensure the interaction safety. In fact, it reduces the risk of harmful collisions by assessing possible danger and it determines interaction capabilities (e.g. for object handover).

The visibility costmap, depicted in Fig. 2b, is computed from the direction of the gaze g and the vector d joining the camera to the point p to observe as follows:

$$c_{visib}(h, p) = \frac{1}{2} \left(\arccos \left(\frac{g}{\|g\|} \cdot \frac{d}{\|d\|} \right) + 1 \right) \quad (3)$$

The gaze direction g and the vector d are computed from the kinematic model h of the human or of the robot.

The visibility costmap models the attention and field of view of the human; it contains high cost for regions of the workspace that are hardly visible by the human. When accounted by the path planner it aims to limit the effect of surprise as a human may experience unease while the robot moves in hidden parts of the workspace. It also provides information about the visibility of objects and the attentional state of the human.

Both distance and field of view constraints are combined and accounted by the path planner and the attentional executive system. The path planner is so able to avoid high cost regions by maximizing the clearance and increasing the robot visibility. The executive system, instead, influences the arm

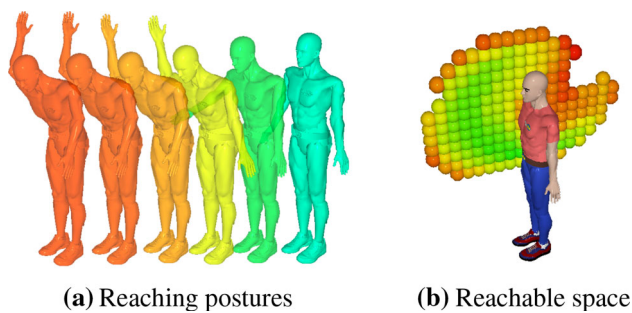


Fig. 3 Reaching postures (a) and a resulting slice of the Reachable space (b) of the right arm. The comfort cost, depicted using *different colors*, is used to model reaching capabilities of the human. (Color figure online)

controller at run-time to modulate the velocity along the trajectory, even stopping the motion when the cost exceeds a certain threshold.

The *reachability costmap*, depicted in Fig. 3b, estimates the reachability cost for a point p in the human or robot workspace. The assumed reachable volume of the human or robot can be pre-computed using generalized inverse kinematics. For each point inside the reachable volume of the human, the determined configuration of the torso remains as close as possible to a given resting position. A comfort cost is assigned to each position through a predictive model of human posture introduced in [31] using a combination of the three following functions:

- The first function computes a joint angle distance from a resting posture q^0 to the actual posture q of the human (see Fig. 3a), N is the number of joint and w_i are weights:

$$f_1 = \sum_{i=1}^N w_i (q_i - q_i^0)^2 \quad (4)$$

- The second considers the potential energy of the arm, which is defined by the difference between the arm and the forearm heights with those of a resting posture (Δz_i) pondered by an estimation of the arm and the forearm weights $m_i g$:

$$f_2 = \sum_{i=1}^2 (m_i g)^2 (\Delta z_i)^2 \quad (5)$$

- The third penalizes configurations close to joint limits. To each joint corresponds a minimum and a maximum limit and the distance to the closest limit (Δq_i) is taken into account in the cost function as follows with a weight γ_i :

$$f_3 = \sum_{i=1}^N \gamma_i \Delta q_i^2 \quad (6)$$

The cost functions are summed to create the reachability cost with the function $GIK(h, p) \rightarrow q$ that generates a fully specified configuration using generalized inverse kinematics:

$$c_{reach}(h, p) = \sum_{i=1}^3 w_i f_i(GIK(h, p)) \quad (7)$$

where h is the human model and w_i weighting the three functions. The musculoskeletal costmap (i.e. the predictive human like posture costmap) accounts for the reaching capabilities of the human in the workspace. It is used to compute object transfer points and, during the path planning for the handover task, to facilitate the exchange of the object at any time during motion such as introduced in [30]. A similar costmap defined for the robot is used by the attentional system to assess the capacity of reaching an object in the workspace.

Apart from the costmaps, the spatial reasoning system provides a large set of data to the attentional system. Such data are the objects position and velocity (pos_o and vel_o where o is the object identifier), the state of the gripper (open or closed), and the distance between the gripper and a given object (d_{go}).

2.2 Attentional Executive System

In a HRI domain, an attentional system should supervise and orchestrate the human–robot interactions insuring safety, effectiveness, and naturalness. Here, simple handover activities are designed using a BBA endowed with bottom-up attentional allocation strategies suitable for monitoring and regulating human–robot interactive manipulation [14,41]. Starting from values obtained from the costmaps, the environment, and the internal states of the robot, the attentional system is able to focus on salient external stimuli by regulating the frequency of sensory processing. It is also able to monitor and orchestrate relevant activities by modulating the activations of the behaviors.

We assume a frequency-based model of attention allocation [15], where the frequency of the sensors sampling rate is interpreted as a degree of attention towards a process: the higher the sampling rate, the higher the resolution at which a process is monitored and controlled. This adaptive frequency provides a simple and implicit mechanism for both behavior orchestration and prioritization. In particular, depending on the disposition and the attitude of a person in the environment, the behaviors sampling rates and activations are increased or decreased changing the overall attentional state of the system. This attentional state can influence the executive system in the choice of the activities to be executed, indeed, high-frequency behaviors are associated with activities with a high priority.

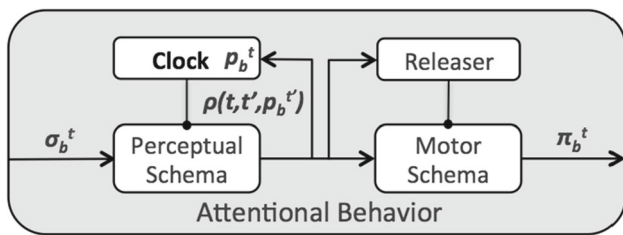


Fig. 4 Schema theory representation of an attentional behavior

2.2.1 Attentional Model

Our attentional system is obtained as a reactive behavior-based system where each behavior is endowed with an attentional mechanism. We assume a discrete time model, with the control cycle of the attentional system as the time unit.

The model of our frequency-based attentional behavior is represented in Fig. 4 by a *Schema Theory* representation [3]. This is characterized by: a *Perceptual Schema*, which takes as input the sensory data σ_b^t (represented as a vector of n sensory inputs); a *Motor Schema*, producing the pattern of motor actions π_b^t (represented as a vector of m motor outputs); a *Releaser* [46] that works as a trigger for the motor schema activation; an attention control mechanism based on a *Clock* regulating sensors sampling rate and behaviors activations (when enabled). The clock regulation mechanism represents our frequency-based attentional allocation mechanism: it regulates the resolution/frequency at which a behavior is monitored and controlled.

This attentional mechanism is characterized by:

- An activation period p_b^t ranging in an interval $[p_{b_min}, p_{b_max}]$, where b is the behavior identifier. It defines the sensors sampling rate at time t . A specific value x for the period p_b^t implies that the behavior b perceptual schema is active every x control cycles.
- A *monitoring function* $f_b(\sigma_b^t, p_b^{t'}) : \mathbb{R}^n \rightarrow \mathbb{R}$ that adjusts the current clock period p_b^t . Here σ_b^t is the perceptual input of the behavior b , t' is the time value at the previous activation, while $p_b^{t'}$ is the period at the previous control cycle.
- A *normalization function* $\phi(f_b) : \mathbb{R} \rightarrow \mathbb{N}$ that maps the values returned by f_b into the allowed range $[p_{b_min}, p_{b_max}]$:

$$\phi(x) = \begin{cases} p_{b_max}, & \text{if } x \geq p_{b_max} \\ \lfloor x \rfloor, & \text{if } p_{b_min} < x < p_{b_max} \\ p_{b_min}, & \text{if } x \leq p_{b_min} \end{cases} \quad (8)$$

- Finally, a *trigger function* $\rho(t, t', p_b^{t'})$, which enables the perceptual elaboration of the input data σ_b^t with a latency period $p_b^{t'}$:

$$\rho(t, t', p_b^{t'}) = \begin{cases} 1, & \text{if } t - t' = p_b^{t'} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

The clock period at time t is regulated as follows:

$$p_b^t = \rho(t, t', p_b^{t'})\phi(f_b(\sigma_b^t, p_b^{t'})) + (1 - \rho(t, t', p_b^{t'}))p_b^{t'} \quad (10)$$

That is, if the behavior is disabled, the clock period remains unchanged, i.e., $p_b^t = p_b^{t'}$; otherwise, when the trigger function returns 1, the behavior is activated and the clock period changes according to $\phi(f_b)$.

2.2.2 Attentional Architecture

The proposed attentional architecture integrates the tasks for pick, place, give, and receive. It is depicted in Fig. 5, where each task is controlled by an attentional behavior. It is also endowed with behaviors for searching and tracking (humans and objects) and with the behavior associated with the obstacle avoidance capability. Each behavior b is endowed with a distinct adaptive clock period p_b^t characterized by its own updating function. In the following, we use the notation $\sigma_b^t[i]$ to refer to the i -th component of the sensory input vector σ_b^t .

SEARCH provides an attentional visual scan of the environment looking for humans. The monitored input signal is $c_{dist}(r, p)$, which represents the distance of the human pelvis p from the robot r in a robot centric costmap (i.e., the input data vector for this behavior is $\sigma_{sr}^t = \langle c_{dist}(r, p) \rangle$). This behavior is always active and it has a constant activation period ($p_{sr}^t = p_{sr}^{t'}$), hence $f_{sr}(\sigma_{sr}^t, p_{sr}^{t'}) = p_{sr}^{t'}$.

Once a human is detected in the robot *far workspace* (i.e., when $3m < c_{dist}(r, p) \leq 5m$), **TRACK** is enabled and allows the robot to monitor the humans motions before they enter in the *interaction space* ($1m < c_{dist}(r, p) \leq 3m$). Also in this case, the monitored signal is the robot-human distance (i.e., $\sigma_{tr}^t = \langle c_{dist}(r, p) \rangle$). In this context, a human that moves fast and in the direction of the robot needs to be carefully monitored (at high frequency), while a human that moves slowly and far away can be monitored in a more relaxed manner (at low frequency). Therefore, the clock period associated with this behavior is updated following the equation (10) with:

$$f_{tr}(\sigma_{tr}^t, p_{tr}^{t'}) = \beta_{tr}\sigma_{tr}^t[1] \cdot \gamma_{tr} \left(\frac{\sigma_{tr}^t[1] - \sigma_{tr}^{t'}[1]}{p_{tr}^{t'}} \right) + \delta_{tr}. \quad (11)$$

Here, the period update is affected by the human position with respect to the robot and the perceived human velocity. In particular, the period is directly proportional to the human distance and modulated by the perceived velocity. The latter is computed as the incremental ratio of the space displacement with respect to the sampling period. The behavior para-

Fig. 5 Behavior-based attentional architecture within the overall framework. The attentional system is provided by the spatial reasoning module with preprocessed data and influences task switching (executive) and motion control (arm controller)

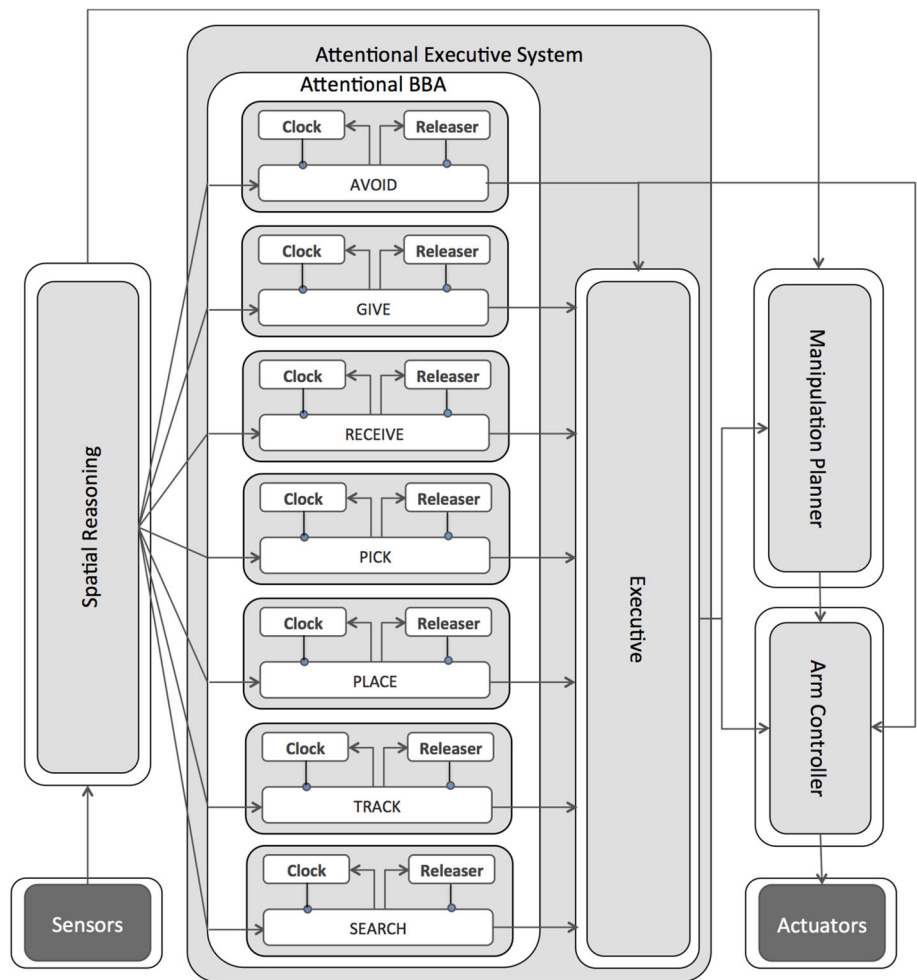


Table 1 Attentional system set up used in the experiments

Attentional BBA	
SEARCH & TRACK	$p_{sr} = 10 \quad \beta_{tr} = 4.5$ $\gamma_{tr} = 0.33 \quad \delta_{tr} = -11.5$
AVOID	$\beta_{av} = 2.01 \quad \gamma_{av} = 1.08$ $\delta_{av} = 0.33 \quad \lambda_{av} = 1.67$
GIVE & RECEIVE	$\beta_{gv} = 0.8 \quad \beta_{rc} = 0.75$
PICK & PLACE	$dmax_{pk} = 0.7m \quad dmax_{pl} = 0.7m$
Costmap thresholds	
Visib. & Reach.	$K_{visibility} = 0.5 \quad K_{reachability} = 0.5$
Executive System	
Task Switcher	$K_{New,Old} = 3$

meters β_{tr} , γ_{tr} and δ_{tr} are used to weight the importance of the human position and velocity in the attentional model and to scale the sampling period within the allowed range. In this specific application the values of these parameters are chosen experimentally (see Sect. 3.1.1 and Table 1), but they

can also be tuned by learning mechanisms either off-line or on-line as shown in previous works [12, 18].

AVOID supervises the human safety during human–robot interaction. It monitors the humans in the *interaction* and *proximity* space and modulates the arm motion speed with respect to the humans’ positions and movements. Moreover, it interrupts the arm motion whenever a situation is assessed as dangerous for the humans. Specifically, the input vector for **AVOID** is $\sigma_{av}^t = \langle c_{dist}(r, p), c_{dist}(h, r), c_{visib}(h, r) \rangle$ representing, respectively, the operator proximity (distance of the human pelvis from the robot base), the minimal distance of the robot from the human body (including hands, head, legs, etc.), and the robot visibility. The human–robot distance $\sigma_{av}^t[1]$ is monitored in the range $0.1m < \sigma_{av}^t[1] \leq 3m$ and **AVOID** is enabled when a human is detected in such an area. If a human gets closer to the robot, then the costs $\sigma_{av}^t[1]$ and $\sigma_{av}^t[2]$ increase and the clock should be accelerated. Instead, the clock should be decelerated, if the operator moves away from the robot. This is captured by the following monitoring function.

$$f_{av}(\sigma_{av}^t, p_{av}^t) = (\beta_{av}\sigma_{av}^t[1] + \gamma_{av}\sigma_{av}^t[2]) \cdot \delta_{av} \left(\frac{(\sigma_{av}^t[1] - \sigma_{av}^t[1])}{p_{av}^t} \right) + \lambda_{av}. \quad (12)$$

In this case, the clock period is directly proportional to the human position $\sigma_{av}^t[1]$ and human–robot minimal distance $\sigma_{av}^t[2]$, while it is modulated by the perceived human speed (with respect to the robot base). Analogously to the previous cases, these components are weighted and scaled by suitable parameters. δ_{av} is thus used to emphasize the period reduction when the human moves towards the robot and, similarly, to increase the period relaxation when the human moves away from the robot base. The β_{av} , γ_{av} and λ_{av} values are chosen as shown in Table 1 in order to weight the importance of the parameters and to scale the period value within the allowed range.

The output of this behavior is a speed deceleration associated with high frequencies. This is obtained by regulating the function $\alpha(t)$ that permits a reactive adaptation of the robot arm velocity (see Sect. 2.3.4). Specifically, $\alpha(t)$ represents the percentage of the speed applied on-line with respect to the one planned. In our case, $\alpha(t)$ is regulated as follows:

$$\alpha(t) = \begin{cases} \frac{p_{av}^t}{p_{av_max}^t}, & \text{if } (\sigma_{av}^t[1] > 0.1m) \text{ and } (\sigma_{av}^t[3] < K_{visibility}) \\ 0, & \text{if } (\sigma_{av}^t[1] < 0.1m) \text{ or } (\sigma_{av}^t[3] \geq K_{visibility}) \end{cases} \quad (13)$$

where, p_{av}^t and p_{av_max} are, respectively, the current activation rate and the maximum allowed period for **AVOID**. Here, if the human is not in the robot proximity and the robot is in the human’s field of view (visibility cost below a suitable threshold, $\sigma_{av}^t[3] < K_{visibility}$), then the velocity is proportional to the clock period (i.e., slow at high frequencies and fast at low frequencies). Instead, if the robot is not visible enough or the human is in the robot proximity, then **AVOID** stops the robot by imposing zero velocity.

PICK is activated when the robot is not holding an object, but there exists a reachable object in the robot *interaction* and *proximity space*. This behavior monitors the distance d_{go} of the target object from the end effector and the associated reachability cost $c_{reach}(r, o)$ (i.e., the input vector for this behavior is $\sigma_{pk}^t = \langle d_{go}, c_{reach}(r, o) \rangle$). Specifically, **PICK** is activated when the distance of the object from the end effector is below a specific distance ($\sigma_{pk}^t[1] \leq 3m$) and the reachability cost is below a suitable threshold ($\sigma_{pk}^t[2] < K_{reachability}$). If this the case, then the associated period p_{pk}^t is updated with the equation (10) by means of the following monitoring function:

$$f_{pk}(\sigma_{pk}^t, p_{pk}^t) = (p_{pk_max} - p_{pk_min}) \frac{\sigma_{pk}^t[1]}{d_{max_pk}} + p_{pk_min} \quad (14)$$

where, p_{pk_min} and p_{pk_max} are, respectively, the minimum and the maximum allowed value for p_{pk} , while d_{max_pk} is the maximum allowed distance between the end effector and the object (refer to Table 1 for the parameters values). This scaling function is used to linearly scale and map $\sigma_{pk}^t[1]$ in the allowed range of periods $[p_{pk_min}, p_{pk_max}]$.

Analogously to the previous case, the speed modulation associated with this behavior is directly proportional to the clock period:

$$\alpha(t) = \frac{p_{pk}^t}{p_{pk_max}} \quad (15)$$

That is, if **PICK** is the only active behavior, then the arm should move with *max_speed* when there is free space for movements (and a low monitoring frequency). Conversely, the arm should smoothly reduce its speed to a minimum value in the proximity of objects and obstacles when precision motion is needed at higher monitoring frequency (this effect is analogous to the one provided by the Fitts’s law [21]).

Once selected by the executive system (see Sect. 2.2.3), the execution of **PICK** is associated with a set of processes: a planning process generates a trajectory towards the given object; upon the successful execution of this trajectory, a grasping procedure follows; finally, if the robot holds the object, it moves it towards a safe position, close to the robot body. Notice that, if **PICK** is not enabled by the executive system this sequence of processes is not activated (indeed, the attentional behaviors provide only potential activations, while the actual ones are filtered and selected by the executive module).

PLACE is activated when the robot is holding an object. Once selected by the executive system (i.e., in the absence of humans in the *interaction space*), this behavior activates a set of processes that move the robot end effector towards a target position, place the object and then move the robot arm back to an idle position. Analogously to **PICK**, **PLACE** monitors the distance of the target d_{gt} and the reachability cost $c_{reach}(r, t)$ (i.e., the input vector for this behavior is $\sigma_{pl}^t = \langle d_{gt}, c_{reach}(r, t) \rangle$). The clock period is regulated by a function, which is analogous to the one of **PICK** (14), while the speed modulation follows the equation (15).

GIVE and **RECEIVE** regulate the activities of giving and receiving objects taking into account the positions and movements of humans in the work space along with their reachability and visibility costs.

GIVE monitors: the presence of humans in the *interaction space* ($1 < c_{dist}(r, p) \leq 3m$), the visibility of the end effector ($c_{visib}(h, r) < K_{visibility}$), the distance ($c_{dist}(r, t)$) and reachability of the human hand ($c_{reach}(h, t) < K_{reachability}$), and the presence of an object held by the robot end effector (distance between end effector and object d_{go} below

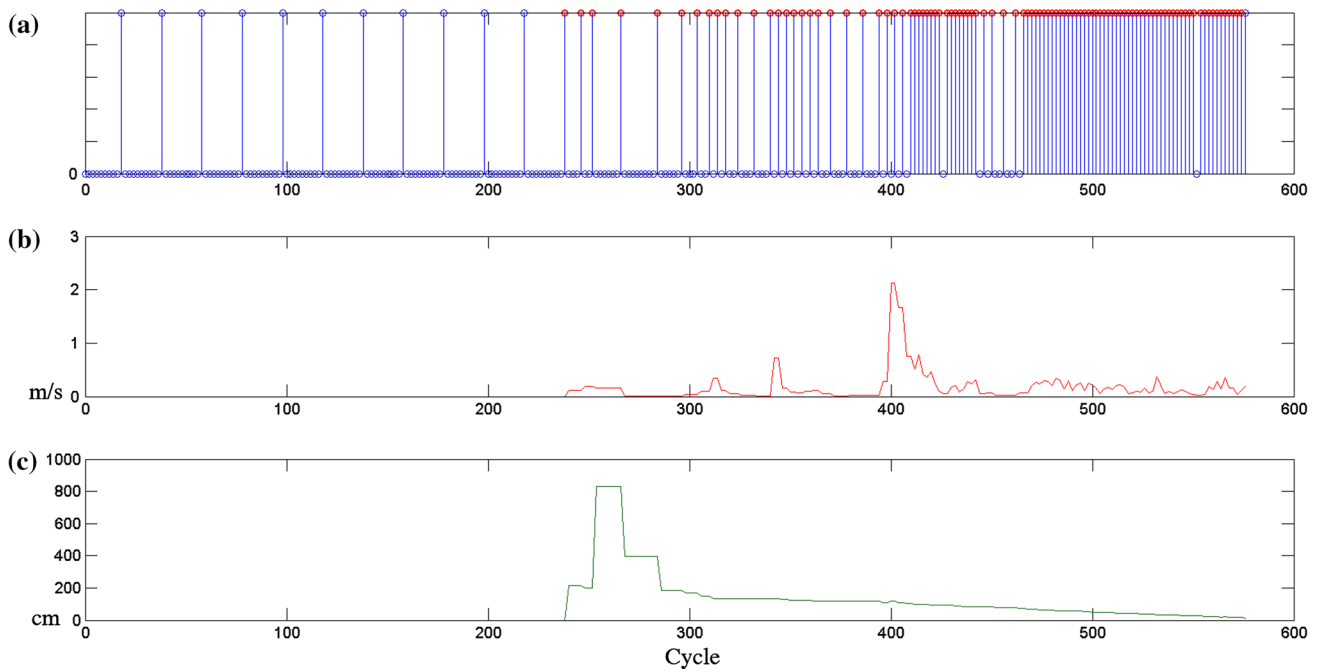


Fig. 6 Execution of GIVE: **a** activations (vertical bars) and releasing (red circles). **b** human hand velocity profile. **c** hand end-effector distance. (Color figure online)

a suitable threshold). That is, the input vector is $\sigma_{gv}^t = \langle c_{dist}(r, p), c_{visib}(h, r), c_{dist}(r, t), c_{reach}(h, t), d_{go} \rangle$.

The clock period is here associated with the distance and the speed of the human hand. If more than one human hand is available, **GIVE** selects the one with a minimal cost in the reachability costmap. Once activated by the executive system, the execution of this behavior moves the end effector towards the target hand; during the execution the robot arm velocity should be regulated with respect to the hand distance and movement. The **GIVE** period changes according to its monitoring function f_{gv} that combines two functions f_{gv}^1 and f_{gv}^2 with a weighted sum regulated by a β_{gv} parameter as follows:

$$f_{gv}(\sigma_{gv}^t, p_{gv}^t) = \beta_{gv} f_{gv}^1(\sigma_{gv}^t[3]) + (1 - \beta_{gv}) f_{gv}^2(\sigma_{gv}^t[3]) \tag{16}$$

The function f_{gv}^1 sets the period proportional to the hand position (i.e. the closer the hand, the higher the sampling frequency) as in equation (14). Instead, f_{gv}^2 depends on the hand speed, that is, the higher the hand speed, the higher is the sampling frequency. The speed of the target hand is calculated as $v = \gamma_{gv} \frac{\sigma_{gv}^t[3] - \sigma_{gv}^t[3]'}{p_{gv}^t}$, where γ_{gv} normalizes the velocity within [0, 1], while the function f_{gv}^2 is used to scale the value of the period within the allowed interval $[p_{gv_min}, p_{gv_max}]$:

$$f_{gv}^2 = \begin{cases} (p_{gv_max} - p_{gv_min})(1 - v) + p_{gv_min} & \text{if } v \leq 1 \\ p_{gv_min} & \text{otherwise} \end{cases} \tag{17}$$

Intuitively, the β_{gv} should be chosen in order to give great priority to the hand position rather than to its velocity (see Table 1), since very quick hand movements are not to be considered as dangerous if the hand is far from the robot operational space. The clock frequency regulates the velocity of the arm movements. More specifically, the execution speed is related to the period and the costs as follows:

$$\alpha(t) = \begin{cases} \frac{p_{gv}^t}{p_{gv_max}}, & \text{if } \sigma_{gv}^t[2] < K_{visibility} \\ -1, & \text{otherwise} \end{cases} \tag{18}$$

In this case, if the human subject is not looking at the robot ($\sigma_{gv}^t[2] \geq K_{visibility}$), then the robot performs a backward movement in the planned trajectory ($\alpha(t) = -1$).

In Fig. 6, we show the activations and releasing activities during the execution of a **GIVE** behavior with respect to the velocity and the distance of a human hand. The **GIVE** motor schema (red circles in Fig. 6a) starts to be active after cycle 230 when the human is in the interaction space and the human hand is reachable ($\sigma_{gv}^t[4] < K_{reachability}$). In this case, it produces a movement towards the human hand. Before that cycle, the perceptual schema is active at low frequency (period = p_{gv_max}) in order to check for the user presence in the interaction space. Around cycle 400, some abrupt movements of the human hand cause an increase of the clock frequency. These effects are attenuated from cycle 450, when the hand stands still. The final high frequency is associated with the object exchange, when the human hand is very close to the robot end effector.

As for **RECEIVE**, this is active when a human enters in the *interaction* space ($c_{dist}(r, p) \leq 3m$) holding an object (distance d_{go} between the object and the end effector less than a suitable threshold), the robot end effector is visible ($c_{visib}(h, r) < K_{visibility}$) and the target human hand is reachable ($c_{reach}(h, t) < K_{reachability}$). Therefore, also in this case the input vector is $\sigma_{rc}^t = \langle c_{dist}(r, p), c_{visib}(h, r), c_{dist}(r, p), c_{reach}(h, t), d_{go} \rangle$. Since this behavior is similar (and inverse) to the one provided by **GIVE**, the sampling rate for **RECEIVE** is regulated by a function which is analogous to the one represented by the equation (16) (set with different parameters) and the adaptive velocity modulation is inversely proportional to the current period, as in equation (18).

2.2.3 Executive Module

The attentional behaviors described so far are monitored and filtered by the executive system, which is to decide about task execution, task switching, and behavior inhibition depending on the current task, the executive/interactive state, and the attentional context. The executive system receives data from the attentional system and manages task execution by orchestrating the human-aware motion planner and the arm movement. In particular, it continuously monitors the active (released) behaviors along with the associated activities (clocks frequencies), and, depending on the current task, it decides: when to switch from one task to another; when to interrupt the task execution; and how to modulate the execution speed.

Initially, the executive system is in an idle state. Once an event activates the attentional behaviors, it can switch from the idle state to one of the following four possible tasks: pick, place, give, and receive. In order to activate a task, the executive system should select not only the associated behavior, but also the most appropriate object for manipulation and the human that should be engaged in the task. Therefore, a task is instantiated by a triple (*behavior, human, object*) and, given a task, we refer to its associated behavior as its *dominant behavior*. Once a task is activated, the executive system should monitor if its dominant behavior remains active during the overall execution. Moreover, it should also decide when to switch to another task if something wrong occurs or a conflict between behaviors is detected (e.g., the activation of **RECEIVE** can conflict with **PICK**, analogously, **GIVE** can conflict with **PLACE**). These conflicts are managed with the following policy: the executive system remains committed with the current task unless the frequency associated with the conflicting behavior exceeds the frequency of the executed one by a suitable threshold: $p_{b_{old}}^t - p_{b_{new}}^t > K_{new,old}$. This simple policy allows us to gradually switch from one task to another if the old dominant behavior gets less excited, while the new one becomes predominant. Notice that this

mechanism allows the robot to keep a stable and predictable behavior reducing also potentially swinging behaviors due to sensors noise. Actually, the swinging behaviors are mitigated not only at the executive level, but also at the behavior-based level. Indeed, even if the system is close to a threshold that can activate/deactivate a releaser due to noise, the behavior activations are gradually increased/decreased avoiding high discontinuity in the attentional state. As an additional mechanism to filter out the outliers, the executive system switches from a task to another only if a repeated indication of this kind is observed. Notice that the target of the task can be switched as well depending on the values of the costmaps (e.g. **GIVE** selects the human hand with minimal reachability values). In our setting the executive system always enables the target suggested by the dominant behavior, however, a thresholding mechanism, analogous to the one for task switching, can be exploited to regulate target commitment.

Furthermore, the executive system monitors the **AVOID** behavior to prevent collisions with objects and humans. Indeed, the arm velocity modulation is obtained as the minimal between the one proposed by the dominant behavior and the one suggested by the **AVOID**: $\alpha(t) = \min(\alpha_{av}(t), \alpha_{task}(t))$. Moreover, **AVOID** can directly bypass the executive system (see Fig. 5) to stop the motion in case of dangerous interactions/manipulations.

2.3 The human-Aware Manipulation Planner

Once a task is selected by the attentional executive system, an associated manipulation task has to be generated by the manipulation planner. The planning process proceeds by first computing a path \mathcal{P} using a “human-aware” path planner [30,43,44], which relies on a grasp planner to compute manipulation configuration and secondly by processing this path using the soft motion generator [10,11] to obtain a trajectory $TR(t)$. In this section we overview the main components of this framework.

2.3.1 Grasp Planner

As the choice of a grasp to grab an object greatly determines the success of the task, we developed a grasp planner module for interactive manipulation [38]. Even for simple tasks like pick and place or pick and give to a human, the choice of the grasp is constrained at least by the initial and final position accessibility and by the grasp stability [6]. The manipulation framework is able to select different grasps depending on the clutter level in the environment (see Fig. 7). Grasp planning basically consists in finding a configuration for the hand(s) or end effector(s) that will allow to pick up an object. In a first stage, we build a grasp list to capture the variety of the possible grasps. It is important that this list doesn't introduce a bias on how the object can be grasped. Then, the planner can

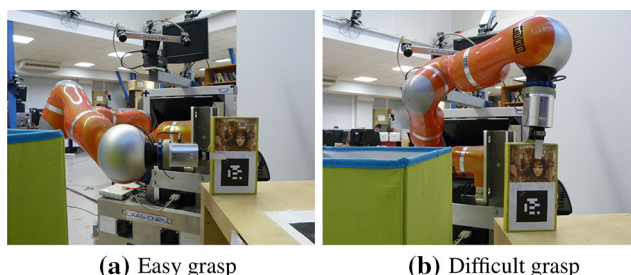


Fig. 7 Ease grasp (a) and difficult grasp (b) depending on obstacles in the workspace

rapidly choose a grasp according with the particular context of the task.

2.3.2 Path Planner

The human-aware path planning framework [30] is based on a sampling-based costmap approach. The framework accounts for the human explicitly by enhancing the robot configuration space with a function that maps each configuration to a cost criterion designed to account for HRI constraints. The planner then looks for low cost paths in the resulting high-dimensional cost space by constructing a tree structure that follows the valleys of the cost landscape. Hence, it is able to find collision free paths in cluttered workspaces (Fig. 10) and account simultaneously for the human presence explicitly.

In order to define the cost function, the robot is assigned a number of points of interest (e.g. the elbow or the end effector). The interest-points positions in the workspace are computed using forward kinematics $FK(q, g_i)$, where q is the robot configuration and g_i the i -th interest-point. The cost of a configuration is then computed by looking up the cost of the N points of interest in the three costmaps presented in Sect. 2.1, and summing them as follows:

$$cost(h, q) = \sum_{i=1}^N \sum_{j=1}^3 w_j c_j(h, FK(q, g_i)) \quad (19)$$

where h is the human posture model, q is the robot configuration and w_i are the weights assigned to the three elementary costmaps c_j of Sect. 2.1. The tuning of those weights can be achieved by inverse optimal control [1], it is out of scope of this paper. When the human is inside the interaction area evaluated by the robot centric distance costmap, planning is performed on the resulting configuration space costmap with T-RRT [26,30], which takes advantage of the performance of two methods. First, it benefits from the exploratory strength of RRT-like planners resulting from their expansion bias toward large Voronoi regions of the configuration space. Additionally, it integrates features of stochastic optimization methods, which apply transition tests to accept or reject

potential states. It makes the search follow valleys and saddle points of the cost landscape in order to compute low-cost solution paths. This human-aware planner outputs solutions that optimize clearance and visibility regarding the human as well handover motions from which it is easy to take the object at all times.

In a smoothing stage, we employ a combination of the shortcut method [5] and of the path perturbation variant described in [30]. In the latter method, a path $\mathcal{P}(s)$ (with $s \in \mathbb{R}^+$) is iteratively deformed by moving a configuration $q_{perturb}$ randomly selected on the path in a direction determined by a random sample q_{rand} . This process creates a deviation from the current path, hoping to find a better path regarding the cost criteria. The path $\mathcal{P}(s)$ computed with the human-aware path planner consists of a set of via points that correspond to robot configurations. Via points are connected by local-paths (straight line segments).

2.3.3 Trajectory Generation

Given the optimized path described by a set of robot configurations $\{q_{init}, q_1, q_2, \dots, q_{target}\}$, the Soft Motion Trajectory Planner [10,11] is used to bound the velocity, the acceleration and the jerk evolutions in order to protect humans. Just as in [42], the trajectory is obtained by smoothing the path at the via points, it is composed for each axis of a series of segments of cubic polynomial curves. The duration of each segment is synchronized for all joints. The trajectory $TR(t)$ obtained is checked for collision and, in case of collision at a smoothed via point, the initial path can be used. In this case the trajectory must stop at the via point.

2.3.4 Reactive Adaptation of the Velocity

To improve the reactivity, the evolution along the trajectory $TR(t)$ is adapted to the environment context using a time scaling function $\tau(t)$; the trajectory realized is then $TR(\tau(t))$. In the absence of human around the robot, it can simply be chosen as $\tau(t) = t$. The function $\tau(t)$ depends on the function $\alpha(t)$ presented in the Sect. 2.2.2.

To maintain dynamic properties of $\tau(t)$, we use the smoothing method introduced in [10]. The function of time $\alpha_s(t)$ represents the smoothed value of $\alpha(t)$. The function $\alpha_s(t)$ is updated at each sampling time (period Δt) of the trajectory controller and directly used to adapt the timing law $\tau(t)$ along the trajectory as follows:

$$\begin{aligned} \tau(0) &= 0 \\ \tau(t) &= \tau(t - \Delta t) + \alpha_s(t) \Delta t \end{aligned} \quad (20)$$

Note that in the case of absence of human, we have $\alpha_s(t) = 1$ and $\tau(t) = t$. The $\alpha_s(t)$ function is analog to the velocity

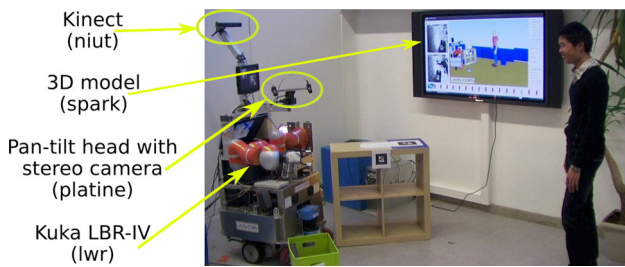


Fig. 8 The Jido platform from LAAS-CNRS

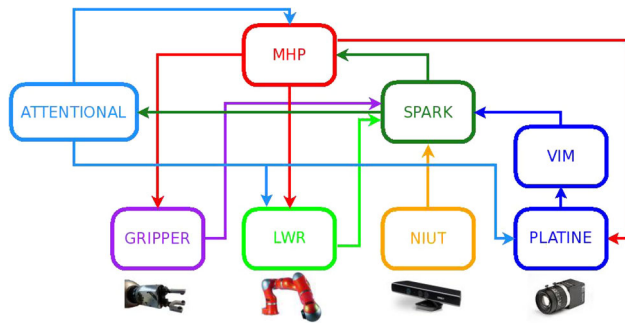


Fig. 9 The main GenoM modules of the software architecture of the Jido Robot

of the time evolution $\tau(t)$. This method adapts the timing law for all joints of the robot that are slowed down synchronously.

In our framework, this mechanism is exploited by the attentional executive system which is able to modulate the speed along the executed trajectory by controlling the parameter $\alpha(t)$ taken as input by the controller.

3 Experiments

In this section, we present a case study along with some preliminary experimental results collected to illustrate the behavior and the performance of the overall HRI system during a typical interaction context (a complete evaluation of the system is left as a future work).

3.1 Setup

To illustrate our approach, we present the results carried out on the LAAS-CNRS robotic platform Jido. Jido is built up with a Neobotix mobile platform MP-L655 (however, mobile robotics tasks are not considered in this paper), and a Kuka LWR-IV arm (see Fig. 8). It is equipped with one pair of stereo cameras and a Kinect is used to track the human body.

The Fig. 9 depicts the main elements of the software architecture of the robot. This architecture is based on GenoM modules [22]. An important module, *Spark*, is responsible for perception and interpretation of the environment combining sensory data and modules results. In particular, it main-

tains the 3D model of the environment tracking positions and velocities of humans and salient objects. A representation of the 3D model is displayed on the large screen in the back of the scene as illustrated in Fig. 8. *Mhp* is the motion planner and *lwr* is the trajectory controller module. *Niut* is in charge of tracking the human kinematics using the Kinect. Using markers, *Viman* identifies and localizes objects while *Platine* controls the orientation of the stereo camera pair. *Attentional* module includes both the Attentional BBA and the Executive.

3.1.1 Parameters Settings

The attentional system parameters have been set as follows. The *far workspace* is in the interval $(3m, 5m]$ meters from the robot base, the *interaction space* is in $(1m, 3m]$, while the *proximity space* is in $[0.1m, 1m]$. For each behavior clock, the period spans the interval $[1, 10]$, while p_{sr} is constant and set at 10. The maximum speed of the human pelvis v_{max} is equal to $3m/s$, while max_speed of the robot end effector is $2m/s$. In TRACK and AVOID, the variable to be tuned are only β_{tr} , β_{av} , and γ_{av} , while γ_{tr} and δ_{av} are about $1/v_{max}$, hence 0.3 (to scale the velocity with respect to its maximum value), instead γ_{tr} and λ_{av} are used to normalize the values within the allowed interval. β_{tr} emphasizes the effect of the human position on the tracking attention, while β_{av} and γ_{av} also regulates the balance between the influence of the $\sigma_{av}[1]$ and $\sigma_{av}[2]$. As for GIVE and RECEIVE, β_{gv} and β_{rc} regulates the importance of velocity and position in the period update. In PICK and PLACE, we set $dmax_{pk} = 0.7m$ and $dmax_{pl} = 0.7m$ because the robot arm extension is about $0.793m$ (kuka lightweight) which is used as a reference to define a maximal distance for targets to be reached. The costmap-related thresholds $K_{visibility}$ and $K_{reachability}$ have been set to 0.5, since the costmap values are normalized in $[0, 1]$ and this setting was natural and satisfactory. Concerning the Executive System, the $K_{New,Old}$ was set to 3 (30% of the maximum allowed period) after manual tuning searching for the best regulation trading off between task commitment (for high values of $K_{New,Old}$ the switch is never enabled) and task switching (for low values of $K_{New,Old}$ the switch is enabled too often). All the parameters values associated with the attentional system have been collected in Table 1.

3.2 Results

Given the setting described above, we tested: the human aware planning system performance in a simplified scenario (a simple pick and give scenario); the attentional system effectiveness in monitoring and controlling activities during tasks of object handover (activation reduction vs. safety and performance); finally, we assessed the overall attentional

Fig. 10 The human aware manipulation planner is able to handle free (*left*) and cluttered (*right*) environments

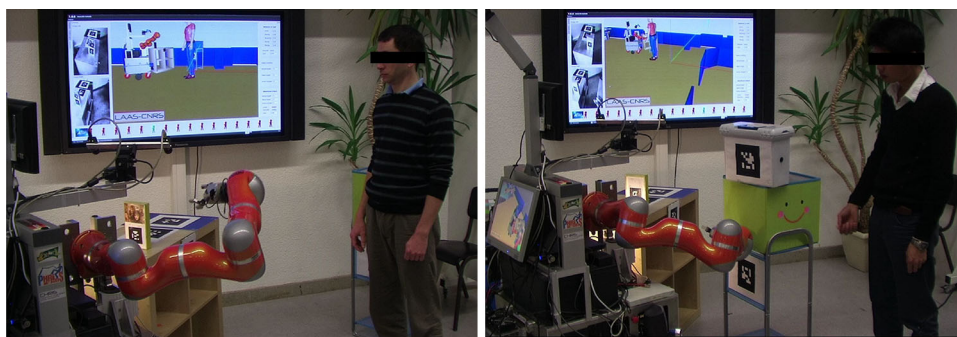


Table 2 Planning and execution performance

Duration	Pick		Give		Total
	Planning	Execution	Planning	Execution	
Mean	1.29 s	6.61 s	2.75 s	10.20 s	20.85 s
Min	0.72 s	5.00 s	0.99 s	5.58 s	12.29 s
Max	5.45 s	24.52 s	12.18 s	22.34 s	64.49 s
STD	0.81 s	3.51 s	2.01 s	3.75 s	5.5 s

system and the way it affects the overall human–robot interaction (quantitative and qualitative analysis).

3.2.1 Human Aware Planning System

In the first experimental test, our aim is to assess the performance of the human-aware planning and control system during pick and give tasks (Fig. 10). With respect to previous implementation of the human-aware planning and control system, the version used here introduces an enhanced T-RRT method to deal with cluttered environments (see Sect. 2.3.2) and a better connection with the controller, which is based on the timing law to regulate the speed (see Sect. 2.3.4).

We assume that the CAD models of the environment are known, while the pose of the objects and obstacles in the environment are updated in real time using the stereo cameras and markers. The position and posture of the humans are updated using the Kinect sensor.

We consider a scenario, where the robot is involved in a pick-and-give task. This task is activated when the following two conditions are verified: there is an object in a reachable position and a human within the robot workspace, who is not holding any objects.

Indeed, as soon as the stereo camera pair detects an object on the table the PICK behavior becomes dominant. Then, once the Kinect detects a human, the GIVE behavior is activated. Both the PICK and GIVE behaviors are associated with planned trajectories generated by the motion planner.

In this experiment, to assess the planner performance we measured: the time to plan the trajectory and the time to execute it for both the pick and the give phases. To verify

the human aware planner capabilities we varied the human and obstacle positions (see Fig. 10a). Table 2 presents the results; these data are the synthesis of 53 trials. Notice that the attentional regulation of speed is here switched off. The visibility and distance property are equally tuned.

The collected data shows that the planning time increases when the environment becomes more cluttered and the trajectory more complex. However, the times obtained with the T-RRT method are compatible with a reactive and natural human–robot interaction when the environment is reasonably uncluttered. For cluttered environment, like the one of the Fig. 10b, the path computed by the planner can become long and complex.

3.2.2 Attentional HRI

In a second experiment, we tested the attentional system by measuring its performance in attentional allocation and action execution. For this purpose, we defined a second, more complex, scenario in which the robot should monitor and orchestrate the following tasks: pick an object from a table, give one object to a human, receive an object from a human or place an object in a basket. In this case, the velocity of the arm is adapted with respect to the positions and the activities of humans in the scene. The robot behavior should be the following. In the absence of a human, the robot should monitor the scene to detect humans and objects. When an object appears on the table, the robot should pick it. In the absence of humans the picked object should be placed in the basket. If a human comes to hand over an object, the robot should receive it (if the robot holds another object, it should

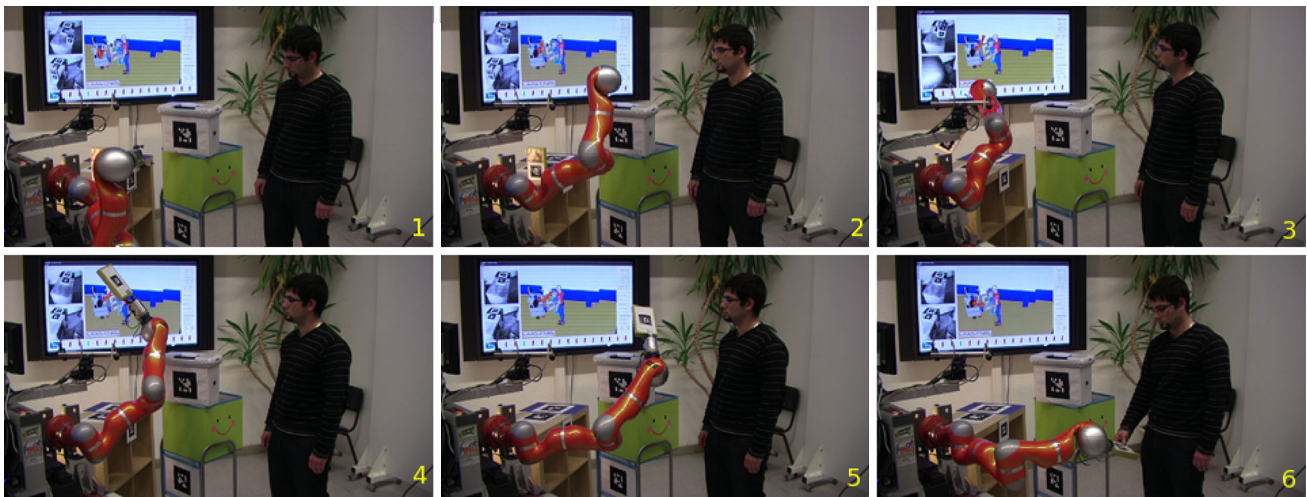


Fig. 11 A complete sequence of pick and give. 1: The robot perceives the human and an object. 2: The robot moves the arm towards the object. 3: Just after grasping the object, the robot starts moving towards the

human. 4: The arm avoids an obstacles. 5: The robot moves the arm towards the human. 6: The human grasps the object handing over by the robot

place it before receiving the new one). If a human is ready to receive an object, the robot should give the object it holds or try to pick an object in order to give it to the human. All these behaviors should be orchestrated, monitored, and regulated by the attentional system. Figure 11 shows a sequence of snapshots representing a pick and give sequence; after picking a tape box on a table, the robot gives it to the human.

Five subjects participated to this experiment: three graduate and two PhD students, two females and three males with an average age of 28. The subjects were not specifically informed about the robot behavior. They were only told that the robot was endowed with certain skills/behaviors such as give or take an object, and that their attitude in the space could somehow have an influence its behavior, but they actually did not know what to expect during the interaction.

In this scenario, we assessed the performance of the attentional system in terms of behavior activations and velocity modulation: the attentional system should focus the behaviors activations on relevant situations only, while the velocity should be reduced only when necessary (e.g., in case of danger, when accuracy is needed or to provide a more natural behavior). To assess the attentional system efficiency in attention allocation, we considered the percentage of behavior activations (with respect to the total number of cycles) and the mean value of the velocity modulation function (represented by $\alpha(t)$ see Sect. 2.3.4) for each interaction phase associated with the execution of a task (i.e., give, receive, pick, place). In particular, for each phase we illustrate the activations of two behaviors: the dominant behavior (i.e., the one characterizing the executed task, e.g., **PICK** during the pick task) and the **AVOID** behavior. The idea is that the attentional system is effective if it can reduce these

activations without affecting the success rate and the safety associated with each phase. Analogously, the mean value of the velocity modulation function $\alpha(t)$ should be maximized preserving success rate, safety, and quality of the interaction. In our setting, activations, velocity, and success rate are measured with quantitative data (log analysis and video evaluation). As for safety and quality of interaction, we collected the subjective evaluation of the testers using a questionnaire, which was compiled after each test session.

The quantitative evaluation results are illustrated in Tables 3 and 4, while the qualitative results can be found in Table 6. The collected data are here the means and standard deviations (STDs) of the 20 trials (4 for each participants) for each phase. Table 3 presents the results obtained by evaluating the logs associated with the trials: we segmented and tagged (comparing them to the corresponding data in the video) each interaction phase (pick, place, give, receive) measuring the associated performance. In this case we measured behavior activations of the dominant behavior (Table 2, first row), the activations of avoid (Table 2, second row) and velocity attenuation $cost(t) = 1 - \alpha(t)$. Instead, in Table 4 we show the duration of the interaction and the system reliability. These data are obtained by evaluating the videos of the recorded tests. In this table, *Time* is for the time needed to achieve the overall task from behavior selection till the success or the failure; *Failures* is for the percentage of failures with respect to the number of attempts. Here, a failure represents any situation in which the task was not accomplished (e.g., robot not able to grasp the object, to give, or to receive the object, wrong selection of place, falling object during execution).

Table 3 Activations and velocity attenuation during different interaction phases (pick, give, place, receive)

	Pick	Give	Place	Receive
Dom.	0.28 ± 0.15	0.18 ± 0.04	0.27 ± 0.09	0.11 ± 0.05
Avoid	0.26 ± 0.12	0.61 ± 0.25	0.31 ± 0.15	0.72 ± 0.25
cost	0.49 ± 0.20	0.62 ± 0.24	0.45 ± 0.17	0.59 ± 0.20

The activation rates of the dominant behavior (Dom.) and of the obstacle avoidance behavior (Avoid) are defined with respect to the total number of cycles. The velocity attenuation ($cost(t) = 1 - \alpha(t)$) represents the percentage of velocity subtracted by the attentional system

Table 4 Duration of the interaction and reliability analysis from video and log evaluation

	Pick	Give	Place	Receive
Time	12.3 ± 6.3s	14.0 ± 1.4s	12.3 ± 3.4s	15.8 ± 6.1s
Failures	10 %	10 %	9 %	20 %

By considering the quantitative results in Tables 3 and 4, we can observe that for each phase, the percentage of the activations of both the dominant behavior and the **AVOID** behavior remains pretty low with respect to the total number of cycles (Table 3), hence the attentional system, as expected, is effective in reducing the number of activations. However, this reduction does not affect the effectiveness of the system performance. Indeed (see *Failures* in Table 4), the system failures remain low for each phases, therefore the attentional system seems effective in focusing the behaviors activations on task/contextual relevant activities for each interaction phase. Indeed, depending on the attentional state of the system some behavior should be more active than others. We recall here that this mechanism not only allows us to save and focus control and computational resources, but also, and more crucially, to orchestrate the execution of concurrent behaviors by distributing resources among them. In our scenario, behaviors involving human interaction have to be frequently activated, but only when this is required. As we expected, during the give and receive phases the number of activations of **AVOID** are greater than the ones for **PICK** and **PLACE**. Indeed, during pick and place, the attentional system should only monitor the presence of humans in the *interaction area*, focusing the activations only in the presence of potentially dangerous situation. As for velocity attenuation (Table 3), the values for $cost(t)$ seem slightly higher during give/receive phases than during pick/place, this is because the interaction with the human needs more caution. In particular, the human hand proximity and movements during the object exchange determine a modulation of the velocity profile. However (as already observed by [20]), if the robot motion is readable for the human, the handover tasks is usually facilitated by the human collaborative behavior, hence the mean value of the velocity attenuation is not that intense. This can also be observed in the time to achieve the goal (Table 4), where the mean durations for the give and receive phases are slightly higher, but

Table 5 HRI questionnaire [1:very bad, 2:bad; 3:inadequate, 4:not enough, 5:almost enough, 6:sufficient, 7:decent, 8:good, 9:very good, 10:excellent]

Section	Question
Personal information	Age?
	Gender?
	How familiarized are you with robotic applications?
General feelings	Safety: Did you feel safe during interaction?
	Naturalness: How did you feel about the naturalness of the interaction?
	Human legibility: Did you understand the robot behavior?
	Robot legibility: Did the robot react accordingly with your behavior?

the slow-down effect of the interaction is not emphasized in a noticeable difference in performance. Here, the human cooperative behavior during the handover seems facilitated by a natural interaction. This is considered by the qualitative evaluation.

The quality of the interaction was assessed by asking the subjects to fill a specific HRI questionnaire after each of the 20 tests. The aim of this questionnaire, inspired by the HRI questionnaire adopted in [19], is to evaluate the naturalness of the interaction from the operator's point of view.

The questionnaire is structured as follows (see Table 5):

- a *personal information* section containing the personal data and the technological competences of the participants. Here, we categorize subjects by their bio-attributes (age, sex), the frequency of computer use and their experience with robotics;
- a *general feelings* section containing questions to assess the perceived intuitiveness of our approach. In order to measure the level of confidence of the human with respect to the interaction, we asked about its safety, naturalness,

Table 6 Qualitative analysis from questionnaire evaluation. For each data we report the associated 0.95 confidence interval

	Pick	Give	Place	Receive
Safety	10 ± 0.00	9.8 ± 0.19	8.2 ± 0.19	7.2 ± 0.35
Naturalness	9.0 ± 0.30	8.6 ± 0.48	8.0 ± 0.30	7.1 ± 1.19
H Legibility	9.8 ± 0.19	9.0 ± 0.30	8.1 ± 0.28	8.0 ± 0.42
R Legibility	9.4 ± 0.23	9.6 ± 0.23	9.3 ± 0.19	6.0 ± 0.52

and about the understanding level with respect to both the human and the robot point of view.

Each entry could be evaluated with a mark from 1 (very bad) to 10 (excellent).

Table 6 presents the results obtained for each interaction phase (pick, place, give, receive); here, *safety*, *naturalness*, *human* and *robot legibility* are means of the marks given by the evaluators. In the table we also report the 0.95 confidence intervals.

By considering results in Tables 4 and 6, we observe that the task is perceived as reliable for each phase, while, as expected, the perceived safety is higher during the pick and give phases (usually the human remains far from the robot during the pick hence this phase is perceived as very safe, while the operation of give is legible for the users), but it is lower during the receive and place phases. In particular, the receive phase is assessed as slightly less natural and this also affects the evaluation of safety (an unnatural behavior is not readable for the human, hence it can be assessed as dangerous). As for the human legibility, for each phase the robot reacts to the human behavior according to the human expectations. On the other hand, from the robot legibility perspective, the robot motion sometimes seems not natural and can be misinterpreted, in particular this happens during receive and place (this affects the perception of safety).

Table 7 illustrates a correlation of qualitative and quantitative results. In particular, we adopted the Pearson correlation index metric for data of Tables 4 and 6. In the table, we also provide the significance of the correlation coefficients (assuming the collected 20 samples for each phase). As expected, we can find an evident inverse correlation between the qualitative and quantitative values, that is, the Time and Failures performances are inversely connected with Safety, Legibility and Naturalness. In particular, we observe for both **GIVE** and **RECEIVE** behaviors a strong correlation for time of execution and safety perceived by participants, and for percentage of failures and human legibility. These correlations are also supported by a satisfactory significance value. The first strong correlation can be explained by the fact that a short execution time is usually associated with reduced activations of the **AVOID** behavior, which is aroused in case of dangerous human positioning or movements. Therefore, when the exe-

Table 7 Correlation (r) and significance correlation coefficient (p) of qualitative (safety, naturalness, human legibility and robot legibility) and quantitative (time and failures) values for GIVE and RECEIVE phases

	Safety	Natural.	H Legib.	R Legib.
GIVE				
Time				
r	-0.78	-0.63	-0.32	-0.25
p	< 0.0001	0.0014	0.0845	0.1438
Failures				
r	-0.56	-0.66	-0.71	-0.46
p	0.0051	0.0007	0.0002	0.0206
RECEIVE				
Time				
r	-0.73	-0.56	-0.78	-0.66
p	0.0001	0.0051	< 0.0001	0.0007
Failures				
r	-0.60	-0.36	-0.75	-0.41
p	0.0025	0.0594	< 0.0001	0.0362

cution is short, it is likely that few dangerous situations have been encountered and the human tester felt safer. The second inverse correlation shows that several failures during the interaction (e.g., end-effector wrong positioning or objects falling) are related to a reduced legibility of the robot behavior for the users. For the **RECEIVE** behavior we have also a strong and significant inverse correlation between Time and Human/Robot Legibility values. Indeed, if the robot is slow in reacting to the human intention of giving an object, the human can experience a difficulty in the interpretation of the robot behavior. This is not observed during the dominance of the **GIVE** behavior because the robot intention of giving something is usually more legible for the interacting human. The other entries of the table provide weaker correlations and less significant values.

Summing up the results in Tables 3, 4 and 6, the attentional system seems effective in attentional allocation, action selection, and velocity modulation (Table 3) while keeping an effective interaction (Table 4) between the human and the robotic system. Moreover, in our case study, the users usually perceived the interaction as safe, reliable, and natural (Table 6).

4 Conclusions

Interactive manipulation is an important and challenging topic in social robotics. This capability requires the robot to continuously monitor and adapt its interactive behavior with respect to the humans' movements and intentions. Moreover, from the human perspective, the robot behavior should also be perceived as natural and legible to allow an effective and safe cooperation with the robot. In this work, we proposed to deploy executive attentional mechanisms to supervise, regulate, and orchestrate the human–robot interactive and social behavior. Our working hypothesis is that these mechanisms can improve not only the interaction safety and effectiveness, but also the behavior readability and naturalness. While visual and joint attentional mechanisms have been already proposed in social robotics as a way to improve the legibility of the robotic behavior and social interaction, here we proposed attentional mechanisms at the core of the executive control for both task selection and continuous sensorimotor regulation.

In this direction, we presented an attentional control architecture suitable for effective and safe collaborative manipulation during the exchange of objects between a human and a social robot. The proposed system integrates a supervisory attentional system with a human aware planner and an arm controller. We deployed frequency-based attentional mechanisms, which are used to regulate attentional allocations and behavior activations with respect to the human activities in the workspace. In this framework, the human behavior is evaluated through costmap based representations. These are shared by the attentional system, the human aware planner, and the trajectory controller to assess HRI requirements like human safety, reachability, interaction comfort, and field of view. In this context, the attentional system exploits the cost assessment to regulate activity monitoring, task selection, and velocity modulation. In particular, the executive system decides attentional switches among tasks, humans, and objects providing a continuous modulation of the robot speed. This dynamic process of attentional task switching and speed modulation should support a flexible, natural, and legible interaction.

We presented a case study used to describe the system at work and to discuss its performance. The collected results illustrate how the attentional control system behaves during typical interactive manipulation scenarios. In particular, our results suggest that, despite the reduction of the behaviors activations, the system is able to keep a safe and effective interaction with the humans. Indeed, the attentional allocation mechanisms seems to suitably focus and orchestrate the robot behaviors according to the human movements and dispositions in the environment. Moreover, from the human perspective, the attentional interaction is perceived as natural and readable. Namely, the attentional system provides

the capability of dynamically trading-off among naturalness, legibility, safety, and effectiveness of the interaction between the human and the robot.

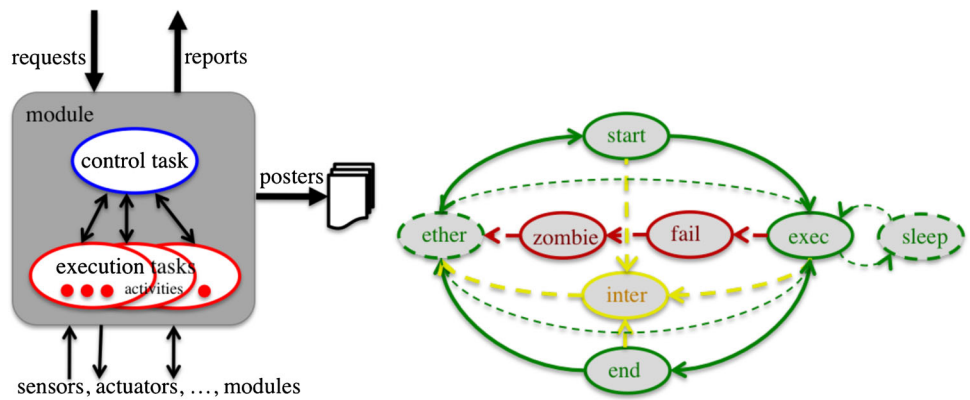
In this work, we mainly focused on the role of the executive attention and attention allocation in simple HRI scenarios, on the other hand we have deliberately neglected other attentional mechanism, which are commonly deployed in social robotics. For instance, a visual attentional system is usually considered as a crucial component that supports a social and natural interaction between the human and the robot [7,8]. These models are complementary with respect to the ones presented in our framework (temporal distribution of attention versus orienting attention in space) and can be easily integrated. For example, in our case study, the SEARCH behavior can be extended by introducing saliency-based methods [25] to monitor and scan the scene. Visual perception is also associated with other important mechanisms for human–robot social interaction and nonverbal communication [29] such as joint attention [28,32,39], anticipatory mechanisms [23], perspective taking [47], etc.. Our behavior-based approach allows us to incrementally introduce analogous models within more sophisticated interaction behaviors to be orchestrated by our attentional framework. For example, we are currently investigating how to integrate more sophisticated human-intention recognition system in our attentional framework [37]. Of course, when the social behavior and the interaction scenario becomes more sophisticated, task-based attentional mechanisms and top-down attentional regulations comes into play [13]. For example, in the presence of complex and structured cooperative tasks [2], the executive switching mechanism should take into account both the behavioral attentional activations (bottom-up) and the interaction schemata required by the task (top-down). The investigation of these issues is left as a future research activity.

Acknowledgments The research leading to these results has been supported by the SAPHARI Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-287513. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

5 Appendix

The overall control architecture has been implemented within the LAAS architecture exploiting the GenoM (Generator of Modules) [22] development framework. In the following, we first introduce the main concepts of the GenoM framework, then we illustrate the implemented control architecture, finally we provide some details about the implementation of the attentional module.

Fig. 12 GenoM module structure and state machine



5.1 GenoM

The GenoM framework allows to design real-time software architectures. It permits to encapsulate the robot functionalities into independent modules, which are responsible for their execution. Each GenoM module can concurrently execute several services, send information to other modules or share data with other modules using data structures called posters. The functionalities are dynamically started, interrupted or parameterized upon asynchronous requests sent to the modules. There are execution and control requests: the first starts an actual service, whereas the latter controls the execution of the services (see Fig. 12). Each request is associated with a final reply that reports how the service has been executed. For each module, the algorithms must be split into several parts: initialization, body, termination, interruption, etc. Each of these elementary pieces of code is called a codel. In the current version of GenoM, these codels are C/C++ functions. A running service is called an activity. The different states of an activity are shown in Fig. 12(right). On any transition, one can go into the INTER state. In case of a problem, one can go into the FAIL state, or even directly into the ZOMBIE state (frozen). Activities can control a physical device (e.g., sensors and actuators), read data produced by other modules (from posters) or produce data. The data can be transferred at the end of the execution through the final reply, or at any time by means of posters.

5.2 System Architecture

A description of the GenoM modules involved in the attentional control cycle is provided in Fig. 13. Here, we can distinguish the SPARK module, which is responsible for perceptual analysis and costmap generation, the MHP module, which is responsible for the robot motion planning and execution (path/grasp/motion planning and smoothing), and the ATTENTIONAL module, which is responsible for attentional regulation and task switching.

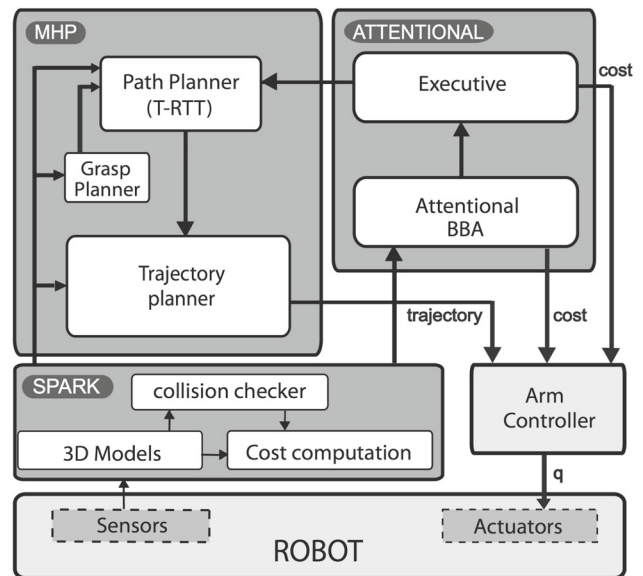


Fig. 13 Architecture of the system

5.3 Attentional System

The attentional system is implemented as a GenoM module that has an executive cycle of 10 milliseconds. An abstract illustration of the codel associated with the attentional system is provided by the Algorithm 1. Here, the *attentionalControlMain()* is activated at each cycle (i.e., every 10 milliseconds) and returns an *ACTIVITY_EVENT* (i.e., the EXEC state). During the cycle, all the behaviors are checked and updated. For each behavior, the attentional module checks if the perceptual schema is active or not. If it is not active, the behavior clock is increased by one tick (*updateClock()*). Otherwise, if the perceptual schema is active, its acts as follows: it reads the associated input data from the poster generated by the SPARK module (*readData()*); it defines the next clock period according to the behavior monitoring function (*updateClockPeriod()*); it assesses the releasing function (*checkReleaser()*) to determine whether the motor schema is active or not; finally,

the previous sensing data is stored (*storeLastSensing()*) and the clock is reset (*resetClock()*). Once each behavior has been updated, the executive system is to select the current activity to be executed and the associated cost (*selectActivity()*).

Algorithm 1 *attentionalControlMain()*

```

for (allBehaviors) do
  if (perceptActive) then
    readData();
    updateClockPeriod();
    checkReleaser();
    if (releaserOn) then
      updateCost();
    end if
    storeLastSensing();
    resetClock();
  else
    updateClock();
  end if
end for
taskAndCost = selectActivity();
reportCycleStatus(taskAndCost);
return EXEC;

```

The executive system is implemented by the *selectActivity()* function (see Algorithm 2). It gets the current executive state (IDLE, PICK, GIVE, RECEIVE, PLACE), the attentional state (active behaviors and the associated periods), and the associated cost vector (velocity modulation suggested by each behavior). If there exists at least one active behavior, the function checks for priorities (depending on the executive state) and decides whether to keep the current activity or to switch to another one. Once one activity has been selected, a target human, location or object is set (*selectTarget()*). Finally, the velocity modulation is decided (*setCost()*) by minimizing the one associated with the selected behavior and the one proposed by AVOID (i.e., $\min(\alpha_{av}(t), \alpha_{task}(t))$).

Algorithm 2 *selectActivity()*

```

getTheExecTask();
getTheAttentionalState();
getTheCostVector();

if (activeBehaviors) then
  priorityEvaluation();
  taskSwitcher();
  selectTarget();
  setCost();
end if
return taskAndCost;

```

Following the standard specifications of a GenoM module, the attentional module is activated by the start function *attentionalControlStart()* (used to initialize the mod-

ule, it returns EXEC) and it is closed by the end function *attentionalControlEnd()* (used to close the module, it returns ETHER).

5.4 Interaction Example

In Fig. 14 we illustrate a sequence diagram that represents a typical pick and give interaction. The diagram shows how the main components of the global framework in Fig. 1 (which is an abstract version of Fig. 13) interacts in the following scenario: the robot picks an object from the table and tries to place it in another position or to give it to a human. For the sake of clarity, we distinguish between an ATTENTIONAL and an EXECUTIVE timeline even though they belong to the same module. On the ATTENTIONAL timeline we show the names of the behaviors whose motor schemas are active (recall that the perceptual schemas of the behaviors are always periodically active). Moreover, to simplify the presentation, only relevant messages are shown. In the absence of a human or when the robot is idling, the robot monitors the scene (*search for human*). The perceptual schema of the SEARCH behavior receives data from the SPARK module (e.g., *no human*). Notice that in Fig. 14, the messages labeled with (*) are periodically transmitted. If an object appears on the table (*object position*), in the absence of other stimuli, the robot tries to pick it up (*pick object*). The EXECUTIVE, as soon as the frequency of (*pick object*) increases, calls the PLANNER for the trajectory generation. Once the planner sends the trajectory to the arm controller, the attentional system should modulate the arm velocity (*speed modulation*) during the execution taking into account the information provided by all the active behaviors. The execution of the trajectory terminates with the object picked (*holding object*). When the robot is holding the object, in the absence of humans, the robot tries to place it on a suitable location (*location position*). The activation of PLACE behavior (*place object*) affects the EXECUTIVE system, which switches to the PLACE mode and invokes the generation of an associated new trajectory (*place trajectory*). During this trajectory execution the attentional system can affect the speed modulation. If a human enters in the INTERACTION_SPACE (*human detected*), TRACK will monitor his/her position (*human position*) and GIVE will be activated (*give object*). In this particular configuration, both PLACE and GIVE behavior are active. The task switcher should choose the one or the other taking into account the frequencies of the two behaviors while monitoring the external processes. If a human is ready to receive an object and the frequency of GIVE becomes dominant, the EXECUTIVE calls a task switch. It stops the execution of PLACE and asks the planner to launch the behavior GIVE (*switch to give*). Once again, during the execution, the attentional system affects the behaviors activations and consequently the arm speed modu-

Fig. 14 Sequence diagram of a typical pick and place/give human–robot interactive activity. Messages labeled with (*) are periodically sent

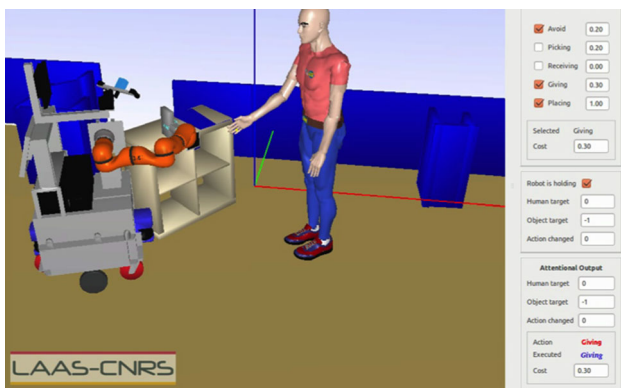
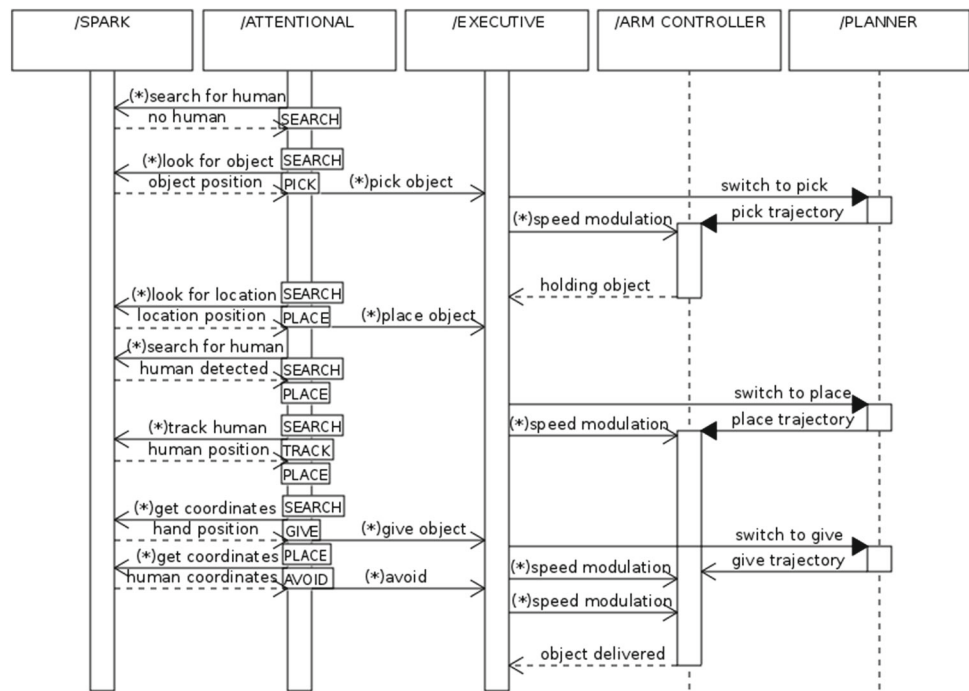


Fig. 15 Snapshot of the interface of the simulated environment, during a typical pick and place/give human–robot interactive activity

lation. In the presence of a human, also the AVOID behavior can give its contribution with speed modulation halting the execution in case of danger.

5.5 Interface

In Fig. 15 we show the interface used to visualize the system behavior. This snapshot shows the case of the parallel activation of PLACE, GIVE and also AVOID behavior presented above. In the right box we can notice, that the active behaviors are these latter three, and that the selected one, under the condition that the *robot is holding* an object, is the GIVE behavior, because there is a man in the scene who is asking for an object.

References

1. Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning, p 1. ACM
2. Alili S, Alami R, Montreuil V (2009) A task planner for an autonomous social robot. In: Distributed autonomous robotic systems. Springer, Berlin, pp 335–344
3. Arbib MA (1998) Schema theory. In: The handbook of brain theory and neural networks. MIT Press, Cambridge, pp 830–834
4. Arkin R (1998) Behavior based robotics. MIT Press, Cambridge
5. Berchtold S, Glavina B (1994) A scalable optimizer for automatically generated manipulator motions. In: IEEE/RSJ Int. Conf. on Intel. Rob. And Sys. IEEE, Munich, Germany
6. Bounab B, Labeled A, Sidobre D (2010) Stochastic optimization-based approach for multifingered grasps synthesis. Robotica 28(07):1021–1032
7. Breazeal C (2002) Designing sociable robots. MIT Press, Cambridge
8. Breazeal C, Kidd CD, Thomaz AL, Hoffman G, Berlin M (2005) Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In: in IROS-2005. ACM/IEEE, Edmon-ton, pp 383–388
9. Brooks RA (1991) A robust layered control system for a mobile robot. In: Iyengar SS, Elfes A (eds) Autonomous mobile robots: control, planning, and architecture (vol 2). IEEE Computer Society Press, Los Alamitos, pp 152–161
10. Broquère X, Sidobre D (2010) From motion planning to trajectory control with bounded jerk for service manipulator robots. In: IEEE Int. Conf. Robot. And Autom. IEEE, Anchorage
11. Broquère X, Sidobre D, Herrera-Aguilar I (2008) Soft motion trajectory planner for service manipulator robot. In: IEEE/RSJ Int. Conf. on Intel. Rob. And Sys. IEEE, Nice, France
12. Burattini E, Finzi A, Rossi S, Staffa M (2010) Attentive monitoring strategies in a behavior-based robotic system: an evolutionary approach. In: Proceedings of the 2010 international conference on

- emerging security technologies, EST '10. IEEE Computer Society, Washington, pp 153–158
13. Burattini E, Finzi A, Rossi S, Staffa M (2011) Cognitive control in cognitive robotics: attentional executive control. In: Proc. of ICAR-2011. IEEE, Tallin, Estonia, pp 359–364
 14. Burattini E, Finzi A, Rossi S, Staffa M (2012) Attentional human-robot interaction in simple manipulation tasks. In: Proc. of HRI-2012, Late-Breaking Reports. ACM/IEEE, Boston
 15. Burattini E, Rossi S (2008) Periodic adaptive activation of behaviors in robotic system. IJPRAI 22(5):987–999 Special Issue on Brain, Vision and Artificial Intelligence
 16. Clodic A, Cao H, Alili S, Montreuil V, Alami R, Chatila R (2009) Shary: a supervision system adapted to human-robot interaction. In: Khatib O, Kumar V, Pappas G (eds) Experimental robotics, springer tracts in advanced robotics, vol 54. Springer, Berlin, pp 229–238. doi:[10.1007/978-3-642-00196-3_27](https://doi.org/10.1007/978-3-642-00196-3_27)
 17. Cooper R, Shallice T (2000) Contention scheduling and the control of routine activities. Cogn Neuropsychol 17:297–338
 18. Di Nocera D, Finzi A, Rossi S, Staffa M (2012) Attentional action selection using reinforcement learning. In: Ziemke T, Balkenius C, Hallam J (eds) From animals to animats 12–12th international conference on simulation of adaptive behavior, SAB 2012, Lecture Notes in Computer Science, vol 7426. Springer, Berlin, pp 371–380
 19. Duguleana M, Barbuceanu FG, Mogan G (2011) Evaluating human-robot interaction during a manipulation experiment conducted in immersive virtual reality. In: Proc. of international conference on virtual and mixed reality: new trends, vol I. Springer, Berlin, pp 164–173
 20. Edsinger A, Kemp CC (2007) Human-robot interaction for cooperative manipulation: Handing objects to one another. In: RO-MAN 2007. IEEE, Jeju, Korea, pp 1167–1172
 21. Fitts P (1954) The information capacity of the human motor system in controlling the amplitude of movement. J Exp Psychol 47(6):381391
 22. Fleury S, Herrb M, Chatila R (1997) Genom: a tool for the specification and the implementation of operating modules in a distributed robot architecture. In: IEEE/RSJ Int. conf. on intel. rob. and sys. IEEE, Grenoble, France
 23. Hoffman G, Breazeal C (2007) Cost-based anticipatory action selection for human-robot fluency. IEEE Trans Robot 23(5): 952–961
 24. Iengo S, Origlia A, Staffa M, Finzi A (2012) Attentional and emotional regulation in human-robot interaction. In: RO-MAN, pp 1135–1140
 25. Itti L, Koch C (2001) Computational modeling of visual attention. Nat Rev Neurosci 2(3):194–203
 26. Jaillet L, Cortés J, Siméon T (2010) Sampling-based path planning on configuration-space costmaps. IEEE Trans Robot 26(4): 635–646
 27. Kahneman D (1973) Attention and effort. Prentice-Hall, Englewood
 28. Kaplan F, Hafner VV (2006) The challenges of joint attention. Interact Stud 7(2):135–169. doi:[10.1075/is.7.2.04kap](https://doi.org/10.1075/is.7.2.04kap)
 29. Lang S, Kleinhagenbrock M, Hohener S, Fritsch J, Fink GA, Sagerer G (2003) Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. In: Proc. int. conf. on multimodal interfaces. ACM, Vancouver, pp 28–35
 30. Mainprice J, Sisbot E, Jaillet L, Cortés J, Siméon T, Alami R (2011) Planning Human-aware motions using a sampling-based costmap planner. In: IEEE int. conf. robot. and autom. IEEE, Shanghai.
 31. Marler R, Rahmatalla S, Shanahan M, Abdel-Malek K (2005) A new discomfort function for optimization-based posture prediction. SAE Technical Paper, Warrendale
 32. Nagai Y, Hosoda K, Morita A, Asada M (2003) A constructive model for the development of joint attention. Connect Sci 15(4):211–229
 33. Norman D, Shallice T (1986) Attention in action: willed and automatic control of behaviour. Conscious Self-Regulation 4:1–18
 34. Pashler H, Johnston J (1998) Attentional limitations in dual-task performance. In: Pashler H (ed) Attention. Psychology Press, East Essex, pp 155–189
 35. Posner M, Snyder C (1975) Attention and cognitive control. In: Information processing and cognition: the loyola symposium. Psychology Pr, Hillsdale, Erlbaum
 36. Posner M, Snyder C, Davidson B (1980) Attention and the detection of signals. J Exp Psychol Gen 109:160–174
 37. Rossi S, Leone E, Fiore M, Finzi A, Cutugno F (2013) An extensible architecture for robust multimodal human-robot communication. In: Proc. of IROS, (2013) IEEE, Tokyo, Japan
 38. Saut JP, Sidobre D (2012) Efficient models for grasp planning with a multi-fingered hand. Robot Auton Syst 60(3):347–357. doi:[10.1016/j.robot.2011.07.019](https://doi.org/10.1016/j.robot.2011.07.019) Autonomous Grasping
 39. Scassellati B (1999) Imitation and mechanisms of joint attention: a developmental structure for building social skills on a humanoid robot. In: Computation for metaphors, analogy and agents, vol 1562. Springer, Berlin, pp 176–195
 40. Senders J (1964), The human operator as a monitor and controller of multidegree of freedom systems. IEEE Trans. on Human Factors in, Electronics, HFE-5 pp 2–6
 41. Siciliano B (2012) Advanced bimanual manipulation: results from the DEXMART project, vol 80. Springer, Heidelberg. doi:[10.1007/978-3-642-29041-1](https://doi.org/10.1007/978-3-642-29041-1)
 42. Sisbot E, Marin-Urias L, Broquère X, Sidobre D, Alami R (2010) Synthesizing robot motions adapted to human presence. Int J Soc Robot 2(3):329–343
 43. Sisbot EA, Alami R (2012) A human-aware manipulation planner. Robot IEEE Trans 28(5):1045–1057
 44. Sisbot EA, Marin-Urias LF, Alami R, Siméon T (2007) Human aware mobile robot motion planner. IEEE Trans Robot 23(5): 874–883
 45. Sisbot EA, Ros R, Alami R (2011) Situation assessment for human-robot interactive object manipulation. In: IEEE RO-MAN. IEEE, IEEE, Atlanta
 46. Tinbergen N (1951) The study of instinct. Oxford University Press, London
 47. Trafton JG, Cassimatis NL, Bugajska MD, Brock DP, Mintz FE, Schultz AC (2005) Enabling effective human-robot interaction using perspective-taking in robots. IEEE Trans Syst Man Cybern 35:460–470
- Xavier Broquère** received his M.Sc. degree in 2007 from Institut Supérieur de Mécanique de Paris, France. He joined the Robotics and InteractionS Group at LAAS-CNRS, Toulouse, France in 2007. He received his Ph.D. degree in 2011 from the Paul Sabatier University with his research on robot trajectory planning in the context of Human-Robot Interaction. Since 2012, he is a software engineer in the Mobile Communication Group at Intel Corporation, Toulouse, France.
- Alberto Finzi** is Assistant Professor at DIETI, University of Naples “Federico II” (Italy). He received his Ph.D degree in Computer Engineering from Sapienza University of Rome (Italy). His research interests include: Cognitive Robotics, Human-Robot Interaction, Executive and Cognitive Control, Autonomous and Adaptive systems, Planning and Scheduling, Multi-agent Systems, V & V methods for autonomous sys-

tems. He has been recently involved in several research projects sponsored by the EC (European Community), NASA (National Aeronautics and Space Administration), ESA (European Space Agency), ASI (Italian Space Agency), FWF (Austrian Science Fund), MIUR (Italian Ministry for University and Research), and private industries.

Jim Mainprice is currently a post-doctoral researcher at the Worcester Institute of Technology, MA. He received his M.Sc. degree from the University of Montpellier II in 2008, and his Ph.D. in Robotics and Computer science from the University of Toulouse in 2012. His research interests are Motion Planning and Human-Robot Interaction.

Silvia Rossi is currently assistant professor at the University of Naples Federico II (department of Electrical Engineering and Information Technologies). She received the M.Sc. degree in Physics from University of Naples Federico II, Italy, in 2001, and the Ph.D. in Information and Communication Technologies from University of Trento, Italy, in 2006. Her research interests include Artificial Intelligence, Multi-agent System and Cognitive Robotics and Human-Robot Interaction.

Daniel Sidobre is currently Assistant Professor at University Toulouse III - Paul Sabatier (UPS) since 1992. Graduated from the Ecole Normale

Supérieure in Cachan in 1981, he received an M.Sc. in mechanics from Pierre et Marie Curie University in 1983, a DEA degree in control from UPS in 1986, a Ph.D. in robotics in 1990 from UPS and the Habilitation a Diriger des Recherches degree from UPS in 2009. He spent one sabbatical year at Mc Gill University, Canada. He teach mechanical and production engineering. He is member of the LAAS-CNRS laboratory, making research on robotic manipulation and human robot interaction.

Mariacarla Staffa is currently Research Fellow at the University of Naples Federico II (Department of Electrical Engineering and Information Technology). She received her B.Sc. and M.Sc. degrees in Computer Science both with honours (cum laude) from the University of Naples Federico II in 2004 and 2008 respectively, and her Ph.D. degree entitled “Attentional Mechanism for Sensory-motor Coordination in Behavior-based Robotic Systems” under the supervision of Professor Bruno Siciliano in 2011. She is member of the PRISMA (Projects of industrial and service robotics, mechatronics and automation) and of the PRISCA (Projects of intelligent robotics and advanced cognitive systems) Laboratories, making research in the fields of Cognitive Robotics, Artificial Intelligence and Human–Robot Interaction.