A Human-Aware Manipulation Planner

Emrah Akin Sisbot, *Member, IEEE*, and Rachid Alami

***Abstract*—With recent advances in safe and compliant hardware and control, robots are close to finding their places in our homes. As the safety barrier between humans and robots is beginning to fade, the necessity to design pertinent robot behavior in human environments is becoming a crucial step. In order to obtain a safe, comfortable, and socially acceptable interaction, the robot should be engineered from top to bottom by considering the presence of the human. In this paper, we present a manipulation planning framework and its implementation human-aware manipulation planner. This planner generates paths not only safe but comfortable and "socially acceptable" as well by reasoning explicitly on human's kinematics, vision field, posture, and preferences. The planner, which is applied into "robot handing over an object" scenarios, breaks the human centric interaction that depends mostly on human effort and allows the robot to take initiative by computing automatically where the interaction takes place, thus decreasing the cognitive weight of interaction on human side.**

***Index Terms*—Geometric reasoning, human–robot interaction (HRI), manipulation, motion planning.**

I. Introduction

THE most important property for a robot is the ability to act and change the environment. A robot perceiving and reasoning would be a computer if it does not act according to its reasoning. A manipulator robot, as its name indicates, is capable of changing its environment and displace and assemble objects with the help of its arm. This ability opens a large field of utility for tasks requiring precision, speed, repeatability, and in some cases for tasks involving a certain amount of danger.

Robot hardware becoming more and more safe and compliant will soon allow robots and humans to work together side by side. The fade of this safety barrier will also introduce manipulator robots in our homes. We are not far from having robots that will help us in our daily lives.

Besides safety, i.e., robot not harming the human even in case of failure in software or hardware components, new notions have appeared to characterize the behavior of the robot [1]. The robot should perform its tasks safely, but also it should behave in a way that its actions are acceptable, legible in terms of intention and comfortable for the human.

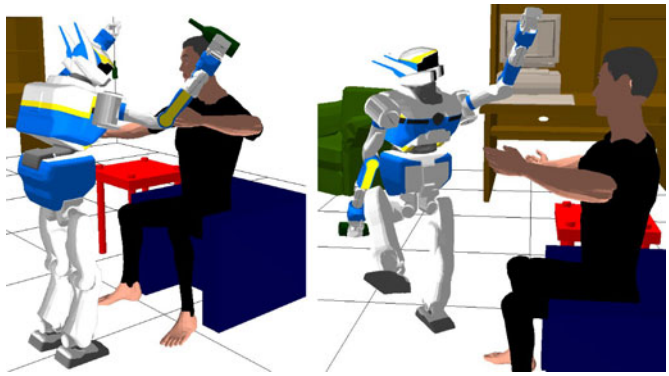


Fig. 1. Humanoid robot HRP-2 hands over a bottle to a sitting person with various object positions and robot placements. (a) Position of the object needs to be chosen according to the comfort of the human. (b) Robot performs a motion that does not show its intention.

These notions are large, somewhat imprecise and subjective, and can be interpreted and tackled at various levels. From user friendly designs [2], [3] to multimodal dialog [4], these notions can be integrated to the robot. Our aim in this work is to study what can be, and what should be done at motion planning and control level in order to have a more systematic and portable way of developing robot tasks and behaviors.

User studies with humans and robots [5]–[8] show a number of new properties and metrics that need to be taken into account in robots behaviors. In this paper, we present a general motion generation framework for a manipulator robot to produce a robot behavior that can be called social in the sense that it tends to respect social rules. We also place ourselves in a "robot handing over an object" scenario and present human-aware manipulation planner that generates robot motions that can be considered acceptable, legible, and comfortable.

Fig. 1 illustrates a scenario where a humanoid robot, HRP-2, hands over a bottle to a person. One of the important aspects of the hand over task is the choice of the place where the object transfer will occur. In order to remove the human's cognitive weight of the interaction, the robot should have the possibility to take the initiative. In the example, this initiative is represented by computing automatically the place of the object transfer. This place needs to be chosen not only by considering robot's accessibility but also human's safety, comfort, and accessibility. Fig. 1(a) illustrates a situation where the robot chooses a "not so comfortable" place to hand over the object.

The legibility is another important issue in all human–robot interaction (HRI) scenarios. When performing a task, the motions of the robot should be legible. The human partner should

Manuscript received May 2, 2011; revised December 9, 2011; accepted April 16, 2012. Date of publication May 9, 2012; date of current version September 28, 2012. This paper was recommended for publication by Associate Editor T. Asfour and Editor G. Oriolo upon evaluation of the reviewers' comments. This work was supported by the E.C. Division Seventh Framework Programme-Information Society Technologies through European Union (EU) CHRIS Project and the EU SAPHARI Project under Contract ICT-215805 and ICT-287513.

E. A. Sisbot was with the Laboratory for Analysis and Architecture of Systems, Centre National de la Recherche Scientifique, F-31077 Toulouse, France. He is now with Toyota InfoTechnology Center, Mountain View, CA 94043 USA (e-mail: sisbot@laas.fr).

R. Alami is with the CNRS; LAAS, F-31077 Toulouse, France, and with Universite de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2012.2196303

understand clearly the intention of the robot without further communication. Fig. 1(b) shows the importance of the legibility with a nonlegible motion. In this example, even if the robot position and objects position will be "good," an unclear motion that does not reflect the robot's intention can surprise human and cause discomfort.

A robot, which behaves in an acceptable manner, should not only consider the feasibility of the task but the safety, comfort, and legibility of its motions as well.

In our previous work [9], we have presented a navigation planning framework in the presence of humans as well as the resulting human-aware navigation planner. This framework was designed for mobile robots navigating while explicitly taking into account the presence of humans. The general mechanism of this framework is based on three cost functions, which are called distance, visibility, and hidden zones, each presenting a different interaction property and related to human's position, orientation, posture, and preferences. With this approach, generated robot paths are nonintrusive, allowing the robot approach humans only if necessary. The resulting robot motions are evaluated safe and comfortable and consider all humans in robot's environment.

The work in this paper is focused on manipulation planning in human–robot close interaction scenarios. This work can be considered as the second step of the human-aware motion framework toward the goal of a complete navigation and manipulation solution in HRI scenarios. Unlike reactive solutions by danger index minimization [10]–[12], we propose a planning framework to the human–robot manipulation problem. Such a planning approach allows us to engage in a motion only if it has been completely planned and validated as well as to avoid to falling into local minima or producing oscillatory motions.

These two motion planning frameworks are also part of a broader effort to develop a decisional framework for human–robot interactive task achievement, embedded in a cognitive architecture, aimed to allow the robot not only to accomplish its tasks but also to produce behaviors that support its commitment vis-a-vis its human partner, as well as to interpret human behaviors and intentions [13].

We have introduced our approach and presented preliminary simulation results in [14] and real-world results in [15]. We have discussed in [6] how user studies have influenced the design of our planner. In this paper, we present in detail a human-aware manipulation framework and an instance of its implementation—human-aware manipulation planner along with its implementation details and results.

Section II provides the main characteristics and algorithms of our manipulation planner. We show simulation results in different scenarios in Section III. Finally, we describe in Section IV the integration of the planner on a mobile robot and present real-world results.

II. Human-Aware Planning

If we think of ourselves when handing over an object to a person, we realize that reasoning about the object, our accessibility, as well as target person's accessibility occur. Before handing over, we have an idea about the rough coordinates of where our hand will reach out and where the person, whom the object is referred, will take the object. These coordinates depend not only on our reach but also on the other person's reach as well as the objects shape and the environment.

In a scenario where a person A hands an object to another person B, we call *object transfer point* (OTP), the spatial point in 3-D workspace where A and B will reach and hold the object together momentarily for its transfer. After the transfer, both hands will retract and the object will be on B's hand.

Various situations may occur in an object hand over scenario where one of the agents anticipates and takes into account other's constraints and/or imposes a choice. In the aforementioned scenario, B may reach for the object before hand or may indicate a point to reach for the object. Here, we are interested in the most difficult case where the person B does not give a hint to contribute the choice of OTP by preplacing his/her hand. The only solution for the person A is to decide and impose a position to place the object.

Therefore, it is very important that a robot handing over an object to a human should not only take into account its kinematic structure and the object but should consider the human part of the task as well. Before beginning its motion, the robot should decide the OTP which needs to be safe and "comfortable" for the human.

Yet, finding a "good" transfer position does not completely solve the problem because the robot's motion to reach that point should also satisfy some conditions. As we mentioned previously, robot should not only move in a safe manner but also needs to ensure the comfort of the human as well as the "legibility" of its motions. In order to ensure the legibility and to make the intentions of the robot sufficiently clear, the handing over motion should be followed by complementary motions based on social rules and protocols (e.g., during the arm motion looking to the object).

To produce comfortable robot motions that take into account all the aspects mentioned previously, a three-stage approach is adopted in human-aware manipulation planner.

1) *Choosing OTP:* The planner finds a safe and comfortable place for the robot to reach out with the object.
2) *Calculating Object Path:* From its current position to OTP, a path for the object is found as it is a free flying body.
3) *Generating Robot Path:* With the object path obtained, the planner finalized the process by generating robot motion that will follow this path.

This decomposition allows us to reduce the complexity of the problem at the cost of the completeness of the planner. With this decomposition, the algorithm has risks of missing a safe and comfortable motion even if it exists. In practical situation, this does not cause a problem because a missing path will probably be too complicated and will require sophisticated motion which can cause the lost of its legibility.

These three stages are run sequentially with the output of one being the input of the next (illustrated in Fig. 2). In case of a failure in one stage, the planner returns to the previous and backtracks with failed object point or object path forbidden. Next sections will explain in detail these stages with underlying HRI notions.

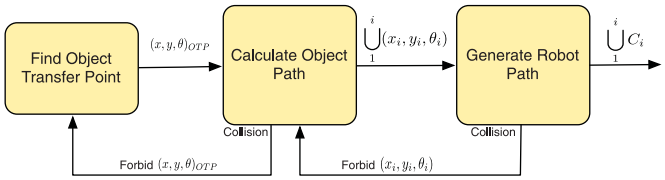


Fig. 2. Human-aware manipulation planner uses a three-stage approach where the output of a block is the input of the next.

A. Object Transfer Point

For a hand over task, one of the key points in the planning is to decide where robot, human, and the object meet. In standard motion planners, this decision is made implicitly by only reasoning about robot's and the object's structure. The absence of human is compensated by letting him adapt himself to the robot's motion, thus, making the duty of the human more important and the motions of the robot less predictable.

User studies [6] and human anatomy allowed us to extract properties that should be taken into account to find a suitable point for object transfer. We have chosen to present these properties with cost functions mapped around the human forming a 3-D grid. Each 3-D coordinate in the near workspace of the human obtains various costs coming from these cost functions.

A grid for human-aware manipulation planner is defined as

$$G = (M_{p,r,s}, H, f_{\text{cost}}, Sp) \tag{1}$$

where $M_{p,r,s}$ is a 3-D matrix containing $p \times r \times s$ cells represented by $a_{i,j,k}$, the cost of the coordinate (i, j, k) in the grid, H is the human, Sp is the resolution, and, finally, f_{cost} represents the cost function attached to the grid. The human representation H contains the kinematic structure of the human, his configuration, as well as his states (e.g., sitting/standing) and preferences (e.g., right/left handed).

The grid is built around the human considering him at its center. Although there is no limit for the dimensions of the grid, for practical reasons, in most scenarios, 2 m $\times$ 2 m $\times$ 1.5 m grids with 0.1-m resolution are used. Each cell in the grid contains a cost that represents and measures an interaction property for the cell's spatial coordinates.

In human-aware manipulation planner, three different interaction properties, which are called "safety," "visibility," and " human arm comfort," are represented as grids with their corresponding cost functions and are used to determine the OTP.

1) Safety: The first of the three properties is the "safety." Ensuring human safety is the absolute need of any HRI scenario. It gains a higher importance in manipulation scenarios where the robot places itself close proximity of the human.

The measure of safety of a 3-D point in human's vicinity is highly related to its distance to the human. Each point around the human can be evaluated in a safety point of view and a cost can be associated with it: The further a point is from a human, the safer it is.

The cost of a point in safety grid represents the measure of safety for the object placed in that particular point. The farther the object is placed from human, the safer the interaction. The safety cost function $f_{\text{Safety}}(H, i, j, k)$ is a decreasing function according to the distance between the human H and object coordinates (i, j, k) in the grid. For example, when the distance between the human and a point in the environment (in the grid) $D(x_i, y_j, z_k)$ is greater than the distance of another point $D(x_k, y_l, z_m)$, we have $f_{\text{Safety}}(H, k, l, m) > f_{\text{Safety}}(H, i, j, k)$. Since the safety concerns lose their importance when the robot is far away from the human, the cost also decreases when getting farther from the human, until some maximal distance at which it becomes null.

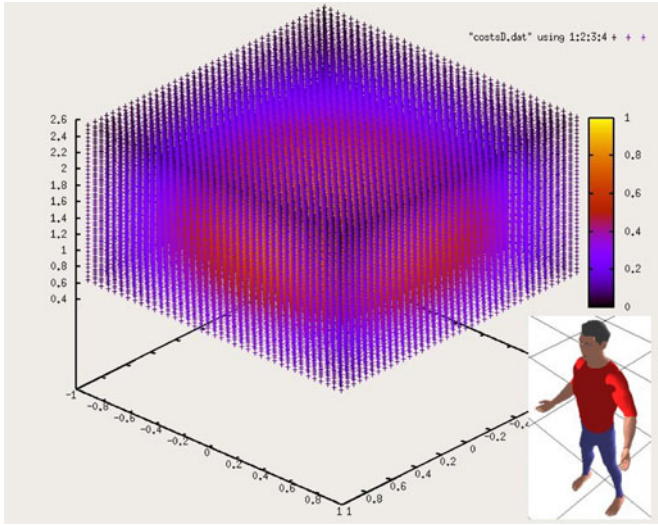


Fig. 3. Costs of safety function mapped around the human. The human is placed at the center of this grid but illustrated at the lower corner for the clarity of the figure. This function creates a protective bubble around the human where costs increase when approaching the person.

As the points in the grids are placed in a 3-D world, the safety risk is evaluated according to human's head and chest which are the most vulnerable parts of the human body. Therefore, the distance between a point and a human is calculated taking the minimum distance between the point, and human head and chest in one axis. This measure can be refined and adapted according to the task and a more detailed model of human vital parts.

The safety function and the safety grid are illustrated in Fig. 3. One should note that the safety const function considers the distance from the center of the object to the human. A more precise computation can be made by evaluating the exact distance between the nearest point of the object mesh model and the human mesh model.

2) Visibility: The visibility of the object is an important property of human–robot manipulation scenarios. The user study conducted by Koay *et al.* [6], where a humanoid robot is assigned to hand over a can to a sitting person, showed that 75% of study subjects preferred the robot to hand over the object in front of them where the object is totally visible. Therefore, the robot has to choose a place for the object where it will be as visible as possible to the human.

We represent this property with a visibility cost function $f_{\text{Visibility}}$. This function alone represents the effort required by the human head and body to get the object in his field of view. If the object is placed directly in front of the human, as the object

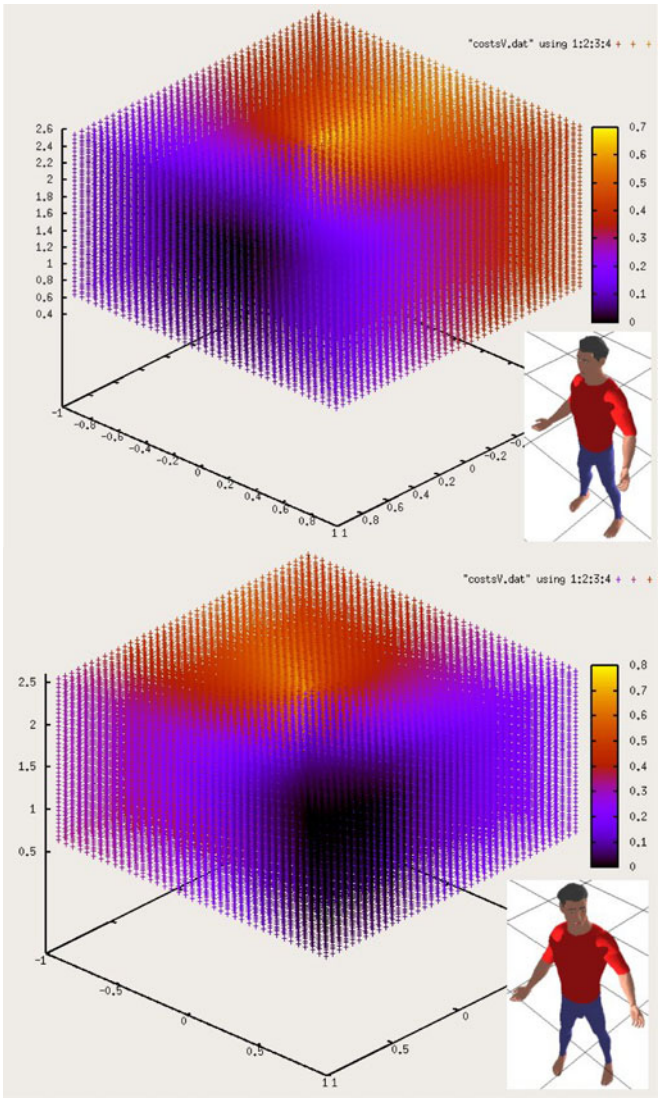


Fig. 4. Costs of visibility function distributed around the human. The human is placed at the center of this grid but illustrated at the lower corner for the clarity of the figure. Points where the human has difficulty to see have higher costs.

is completely visible and no effort is required, the resulting cost of objects placement will be null. On the contrary, when placed behind the human, as in order to see that object the human needs to turn his head and his body, the effort is higher, and thus results in a higher cost.

With a given eye motion tolerance, a point (i, j, k) that has a minimum cost is situated in the cone situated directly in front of human's gaze direction. For this property, the eye tolerance for human as well as any preferences or disabilities that he/she can have are used to compute $f_{\text{Visibility}}$.

With this cost function, the visibility grid is illustrated in Fig. 4. We can see that points at the direction of human's gaze have lower costs. The more the human has to turn his head to see a point, the higher the cost becomes.

3) Arm Comfort: The last property of the placement of the object is the comfort of human's arm configuration when he/she tries to reach to the object. It is a key notion to take into account for a comfortable handing over motion. The robot should reason about human's accessibility and his kinematics to find an OTP which is not only reachable by the human but is comfortable to reach as well.

Human ergonomics [16], [17] and user studies [18], [19] provide very detailed work on the posture of human arm. In [18], Katayama and Hasuura proposed five criteria that play more or less important roles for a human arm posture, and concluded that minimization of muscle stress plays the biggest role in the comfort of an arm. Kang *et al.* [20] showed a natural reach motion that minimizes the work done by the arm.

Although muscle stress and the work are the main properties for the human arm when reaching, they are highly related to arm motion and its load, and it needs a detailed knowledge of the physiology of human. In our case, when we want to evaluate only a posture without any motion, these two properties do really represent the comfort. In Katayama's order of the five criteria, the "medium joint angle index," representing the angular difference in joints for an arm posture, came up as the second most important factor for a comfortable reach. For a comfort-wise evaluation of the arm posture when reaching a point (i, j, k), the cost function $f_{\text{displacement}}$ is defined as

$$f_{\text{displacement}}(H, i, j, k) = \sum_{j=1}^{n} (\theta_{\text{rest},j} - \theta_j)^2 \tag{2}$$

where θ_j is a joint angle of the jth joint, n is the number of arm joints, and θ_{rest} is the angle of the joint in the rest position. This function evaluates the comfort of human arm when reaching to a point in space with measuring the angular change in arm's degrees of freedom (DOF). This definition implies that the displacement cost ($f_{\text{displacement}}$) will be null when the reaching position is the resting position.

Another notion that we will take into account is the comfort related to the potential energy of the arm when performing a reaching motion. It is clear that when reaching an object, it is more comfortable to reach a low point than a high point because when reaching high, the muscles need to support and bring all the weight of the arm to that point. This notion is represented with the cost function $f_{\text{potential}}$ defined as

$$f_{\text{potential}}(H, i, j, k) = \sum_{j=1}^{m} m_j g r_j \tag{3}$$

where m_j is the mass of the jth mass, m is the number of arm masses, and r_j is the coordinates of the center of gravity of jth mass in environment frame.

Both of the cost functions, $f_{\text{displacement}}$ and $f_{\text{potential}}$, receive spatial coordinates of a point and return an evaluation of comfort of the arm's posture when reaching that point. In order to find the arm reaching posture, IKAN [21] algorithm is used. This inverse kinematics solver proved to be very fast and generates ergonomic arm postures.

These two cost functions are merged to one function. As there is no restriction on which hand the human prefers to use,

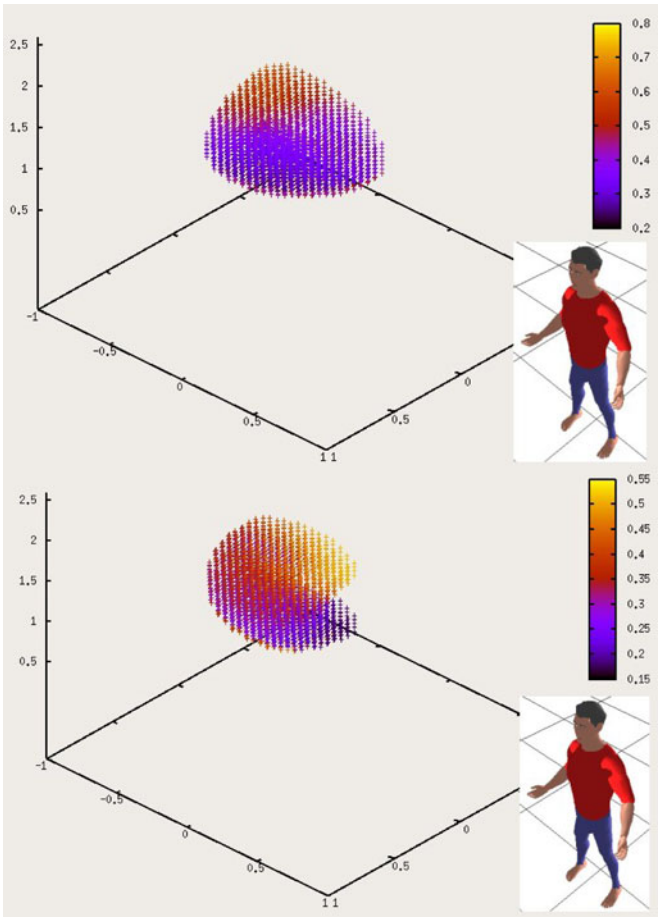


Fig. 5. Arm comfort function for a left-handed person. Although the shapes of left and right arm functions are the same, a penalty is applied to the right arm, thus increasing its costs. Note that only the accessible and more comfortable points are illustrated. Other points around the human have the highest costs in this grid.

separate functions $f_{AC_{L/R}}$ for left and right arm are defined as

$$f_{AC_{L/R}}(H,i,j,k) = \beta_1 f_{\text{displacement}_{L/R}}(H,i,j,k) + \beta_2 f_{\text{potential}_{L/R}}(H,i,j,k) \quad (4)$$

where β_1 and β_2 represent weights that can be given to arm displacement and energy properties.

The separate "arm comfort" cost functions for left and right hands are merged and the final arm comfort cost function, which represents the comfort of human arms represented with $f_{AC}(H,i,j,k)$, is defined as

$$f_{AC}(H,i,j,k) = \min(f_{AC_L}(H,i,j,k) + P_{\text{left}} \; f_{AC_R}(H,i,j,k) + P_{\text{right}}) \quad (5)$$

where P_{left} and P_{right} represent the penalties coming from left/right handedness. For example, for a left-handed person, his preference of reaching to a point would be with his left hand; thus, the cost function will have $P_{\text{left}} < P_{\text{right}}$. The arm comfort functions for left and right arms for "left-handed" person are illustrated in Fig. 5 in the form of a 3-D grid around the human.

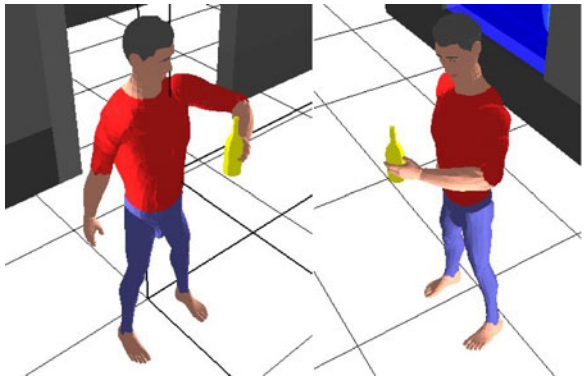


Fig. 6. Object's final placement, i.e., the OTP, is found according to the minimization of safety, visibility, and arm comfort cost functions. In this case, the weights applied to each of these functions are equal.

4) Finding Object Transfer Point: In order to find the OTP, the cost functions mentioned previously are combined to form a single cost function

$$f_{\text{OT}}(H,i,j,k) = w_{\text{Safety}} f_{\text{Safety}}(H,i,j,k) + w_{\text{Visibility}} f_{\text{Visibility}}(H,i,j,k) + w_{\text{ArmComfort}} f_{AC}(H,i,j,k). \quad (6)$$

With the weighted sum of all three functions, the costs in the final function obtain a balance between safety, visibility, and human arm comfort. In order to find the OTP, f_{OT} is mapped around the human to form the object transfer grid, G_{OT}. The cells in this grid are scanned and the cell with the minimum cost is assigned to be the OTP

$$\text{OTP} = ((i,j,k,)|\min_{i,j,k}(f_{\text{OT}}(H,i,j,k))). \quad (7)$$

Fig. 6 illustrates the computed OTP—the place where the robot will carry the object in its hand. This point can be evaluated as safe, visible, and easily accessible to the human. Although this process depends only the human and his characteristics, combining weights of grids need to be chosen carefully. Different choices of these weights result to different OTP's. In Fig. 7, ten points having lowest costs with different weights are illustrated. When combining grids, if $w_{\text{Safety}} > w_{\text{Visibility}} > w_{\text{ArmComfort}}$, the resulting OTP will be as safe as possible (which also means as far as possible), unlike $w_{\text{ArmComfort}} > w_{\text{Visibility}} > w_{\text{Safety}}$ which results the OTP more accessible and comfortable for human's arm. In a real application, one can think that as the familiarity of the human toward the robot increases, the weights can be tuned by the user in order to allow the robot to approach more.

B. Object Path

As the target position of the object is found in the previous stage, in the second step, a path connecting the object's actual position (robot's hand) and final position (OTP) will be found.

The object is considered as a free flying body, and the path found is for the object from its actual position to its final position. In order to compute the path, the object path grid $G_{\text{ObjectPath}}$ is

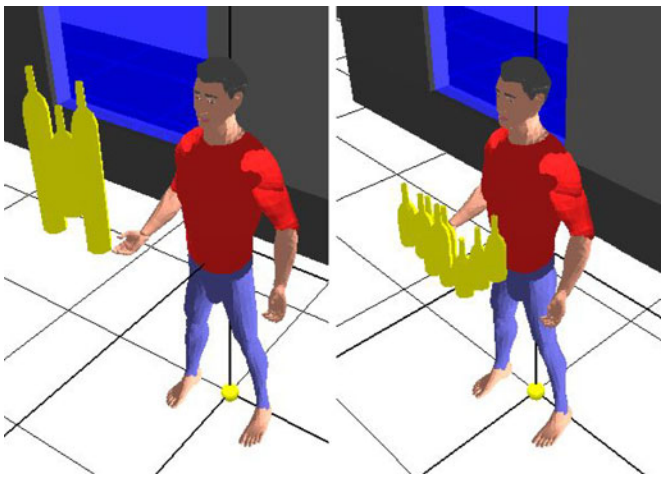


Fig. 7. Ten points around the human having lowest costs of f_{OT}. The order of weights changes drastically the result of the process. The figure on the left illustrates the case where the safety is the most important ($w_{\mathrm{Safety}} > w_{\mathrm{Visibility}} > w_{\mathrm{ArmComfort}}$), unlike the right figure's case, where the arm comfort is dominant ($w_{\mathrm{ArmComfort}} > w_{\mathrm{Visibility}} > w_{\mathrm{Safety}}$).

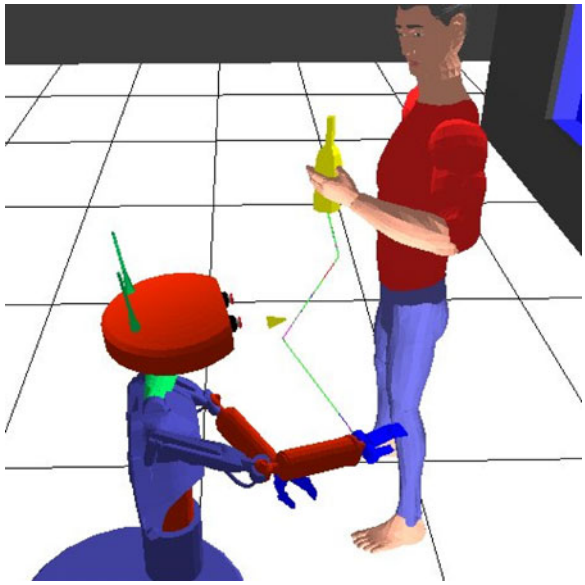


Fig. 8. Object's motion is planned as it is a free flying object. This path is the result of a $A^{\star}$ search that minimizes the combination of safety and visibility cost functions all along object's motion.

built by combining f_{Safety} and $f_{\mathrm{Visibility}}$ cost functions. As the human will not reach the object during its motion, the function f_{AC} is not considered in the process.

With this definition, the object path grid represents a combination of visibility and safety grids. After its construction, a 3-D $A^{\star}$ search with diagonal distance heuristic is used to find a minimum cost path that will be safe and visible at the same time.

At this stage, the obstacles in the near vicinity of the human and the robot are mapped to a grid structure to depict free and occupied space. In order to reduce unnecessary collision tests, the obstacles are inflated by half of their longest dimensions when presented in this grid. This occupancy is taken into account in the $A^{\star}$ search to produce a collision-free object path. There may be two reasons of failure in the path finding process: Either the OTP is in collision, or OTP is in such a constrained space that there is no collision-free path. In both cases, the solution is to regenerate a new OTP. Thus, in the case where no path is found, the grid cell corresponding to the current OTP is tagged as a forbidden cell and the system returns to the first step to generate a new OTP (which comes to picking up the next best OTP from the previously computed OTP list). Fig. 2 illustrates this by tagging the (x, t, θ) coordinates of OTP as forbidden and by returning to *find OTP* step.

The resulting path, which is illustrated in Fig. 8, will be the path that the object and the robot's hand will follow. Obtaining a safe and visible path for the object helps the robot more easily adapt its structure to follow this path and generate a comfortable motion.

Neither the final grid nor three cost grids are constructed explicitly but the values of the cells are calculated for the ones explored during $A^{\star}$ search. This avoids the planner exploring the whole space during its search.

C. Robot Path

Even though we found a path for the object (and robot's hand) to follow, it is not enough to produce an acceptable robot motion in HRI context where the motion should be safe, comfortable, and "legible." With this motion, the robot must make clear its intention.

The third and final stage of planner consists of finding a path for the robot that will follow object's motion. The object's path is computed as it was a free flying object. However, in reality, it is the robot who holds the object and who will make the object follow its path.

To adapt the robot structure to the object's motion, we use generalized inverse kinematics (GIK) [22]–[24] algorithm. Although this method is computationally expensive, it has certain advantages.

1) *Not dependent on the robot structure:* The GIK method only needs a Jacobian matrix that is easily obtainable from robot's structure. This property makes this method easily portable from one robot to another.
2) *Multiple tasks with priorities:* This method allows us to define additional tasks next to the main task. Therefore, the robot not only accomplishes its task but also can take into account additional tasks during its motion. Since the solver considers the priorities of the tasks, any conflicts between joints coming from the presence of multiple tasks are solved automatically by the order of priority without any intervention from the developer.

We use two tasks with different priorities to find an acceptable posture. The first task with higher priority contains the joints that affect the hand of the robot (shoulder, elbow, wrist, and waist). This task aims to reach to a given position in object's path. The second task, with lower priority, controls robot's gaze direction (camera joints) including all the joints that affects robot's head

(waist and neck). The main purpose of this latter task is to increase the legibility of robot's motion by expressing explicitly its intention by looking at the object.

Finally, to generate robot motions that will follow object's path, the object path is divided into samples. The path sampling rate that is used in this stage is equal to the sampling rate of the object path grid. The sampling rate is chosen arbitrarily and according to the scenarios. A higher sampling rate will better represent the object's path at the cost of slowing the planning process.

For each point of the sampled object path, GIK is executed and a robot posture is found to reach to that point. The robot motions between two samples are generated with a linear interpolation of robot configurations. The robot posture is tested against collisions with the obstacles, with itself, and with the whole human body. In case a collision is found, the corresponding grid cell is tagged as forbidden, and the system returns to the previous stage to generate a new object path without considering the forbidden cell. For highly redundant robot kinematics GIK can also be used to avoid the obstacles by integrating a repulsive function as a high-priority task.

With this method, the robot's posture is adapted to object's motion. Even though the first task (motion of robot's arm) is enough to follow the object's path, the supplementary task of moving the head helps the robot express its intention clearly, thus making the motion more legible and the interaction more comfortable.

At the end of this stage, a path is obtained, shown in Fig. 9 for the robot which is safe, visible, and comfortable to the human as we took into account his accessibility, field of view, and his preferences.

The number of tasks can be increased to include more properties or to represent other types of interaction protocols (for example, in the case of a humanoid robot, during the motion of robot's arm and its head, the robot can also point to the object with its other hand to make robot's motion more legible).

III. Simulation Results

In this section, we will illustrate the human-aware manipulation planner in multiple scenarios with various robots. The planner is implemented in Move3D [25] software platform.

Fig. 10 illustrates a scenario where the robot hands over an object to a standing person. The human is looking toward the robot. The robot's base is placed on a position where the human is accessible. The planner calculates a path for the upper body of the robot by ensuring human's safety and comfort. The motion of the robot is easily understandable with its head and arm moving together.

A comparison between a standard motion planner and humanaware planner is illustrated in Figs. 11 and 12 where a humanoid robot hands over a bottle to a right-handed sitting person. In Fig. 11, the robot motion is calculated with a classical motion planner (yet, the OTP is found by the human-aware manipulation planner). As seen in this figure, only the arm of the robot moves. The object and robot's arm enter suddenly in human's field of view with a direct motion and block right

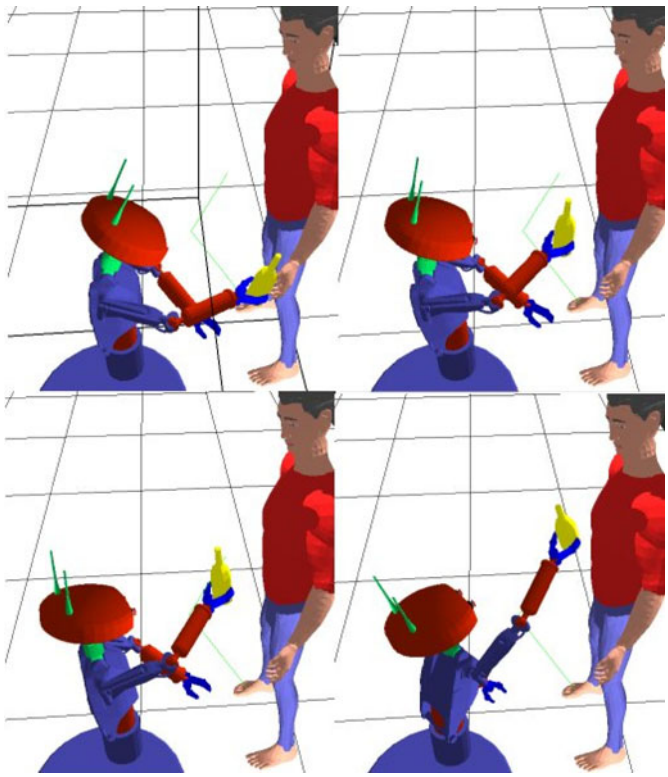


Fig. 9. Calculated path for a "handing over an object" scenario. The robot looks at the object during this motion, ensuring the clarity of its intention to its human partner.

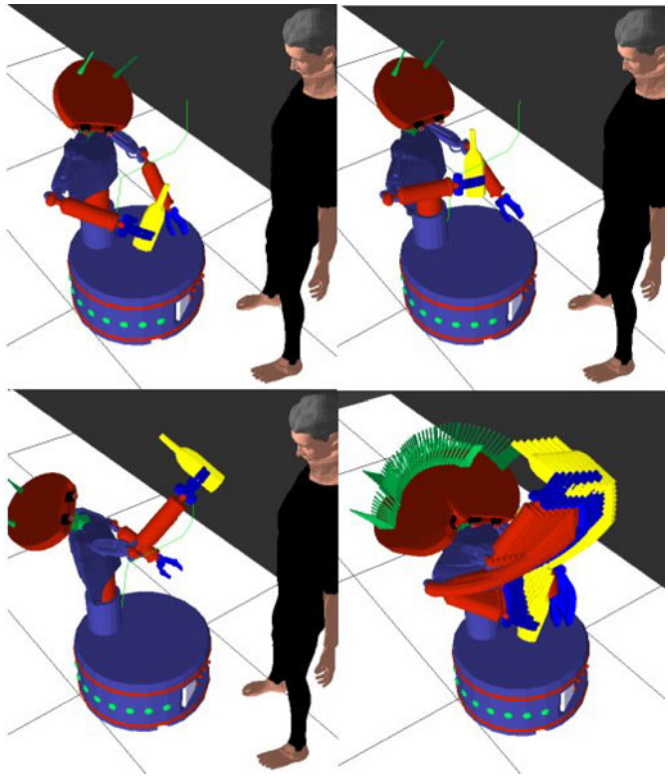


Fig. 10. Path planned by human-aware manipulation planner. The object placement and robot posture are safe and comfortable, and the robot's intention is clearly shown. The right bottom figure illustrates a trace route of the motion of the robot.

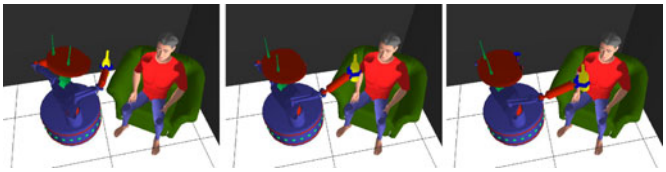


Fig. 11. Path planned by a "classical" motion planner. The final configuration needs to be given explicitly to the system. In these motions, robot arm makes a direct motion toward its goal, appearing suddenly in human's field of view and blocking his right arm.

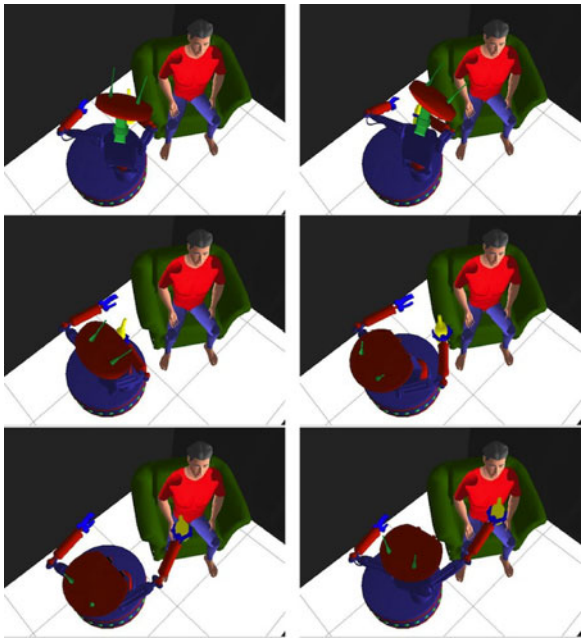


Fig. 12. Scenario where the robot hands over a bottle to a sitting person with human-aware manipulation planner. The object placement and robot posture are safe and comfortable and clearly show the robot's intention.

arm of the human during the movement. The intention of the robot is not easily comprehensible and the motion might create discomfort.

Fig. 12, on the other hand, illustrates the robot path calculated by the human-aware manipulation planner. At the beginning of its motion, the robot looks at the object. It pulls the object toward its body far from the person. When the object enters the human's field of view, the robot begins to bring it forward until reaching the OTP. During its arm motion, the whole upper body of the robot moves, and its head follows the object. This behavior ensures the legibility of robot's motion and can be considered acceptable.

Human-aware manipulation planner depends mainly on the human and not on the robot. The use of GIK allows the planner to be used with different type of robots. Fig. 13 illustrates a scenario with two different robots. The robots are kinematically very different: one being the humanoid robot HRP-2 and one Jido, which is a mobile industrial manipulator. The goal for both of the robots is to give the bottle to the standing person. The point where object transfer will occur is found for both robots the same way because of the fact that it does not rely on robot structure. Even though the structures of the robots

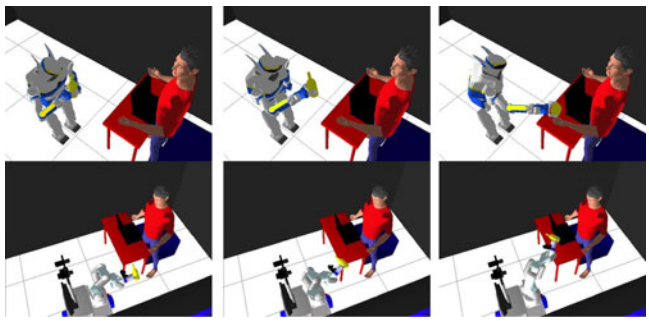


Fig. 13. Planner is easily portable and adaptable to different type of robots. In the top figure, HRP-2 uses HAMP to hand over a bottle to a standing person. In the bottom figure, the same planner is used on Jido.

are significantly different, the planner generates a robot path for both of the robots. For HRP-2, the path generated by the planner takes into account two tasks: Follow object's path, and look at the object.

IV. Robot Integration and Results

The human-aware manipulation planner is integrated to Jido robotic platform in LAAS/CNRS. Jido is a MP-L655 platform from Neobotix, equipped with a Mitsubishi PA-10 arm (with 6 DOF). Several sensors are available on the platform: Sonars, two laser range finders, two stereo camera banks (one mounted on the arm and the other on a pan-tilt unit on the base platform), several contact sensors, and a force sensor on the gripper.

The planner is integrated in LAAS architecture [26] as a Genom [27] module. Fig. 14 illustrates the connection of the planner to the other modules along with the exchanged data. As illustrated in this figure, the planner module is strongly linked with three modules.

1) GEST [28]: This perception module is in charge if detecting, localizing in 3-D, and tracking human head as well as both hands by using the top camera of Jido. The detection of human parts are performed using their shapes, colors, and positions. This module provides the positions of human head and hands.
2) HumPos [29]: HumPos module detects humans through the robot's front laser sensor. This module provides the position and the orientation (based on the motion) of human legs.
3) Xarm [30], [31]: The plans generated by the manipulation planner are executed by this module. Xarm generates trajectories with bounded jerks for Jido's arm and provides arm's actual configuration.

A. Human-Aware Motion Planner as a Genom Module

The planner module works with a static 3-D map along with human's model, his cost function parameters, and the robot model. The necessary environment data (including robot's and human's) are acquired constantly. Fig. 15 illustrates the inner structure, as well as the inputs/outputs of the module.

Grids and GIK solver establish the core of the planner. Human configuration is permanently updated with the 3-D head/hand

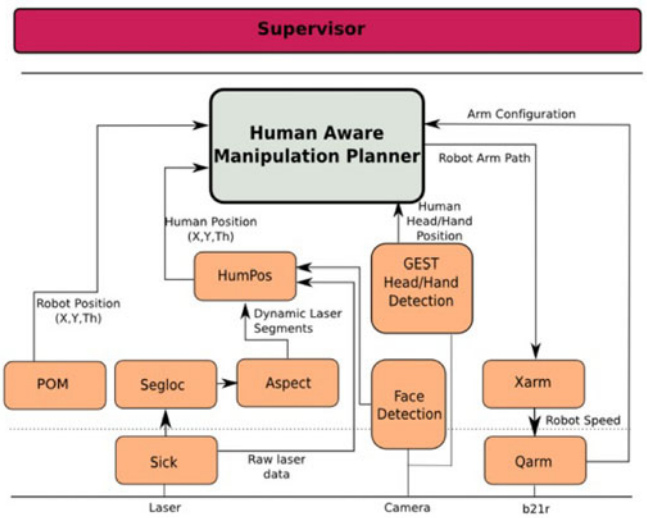


Fig. 14. Integration of the HAMP in LAAS architecture.

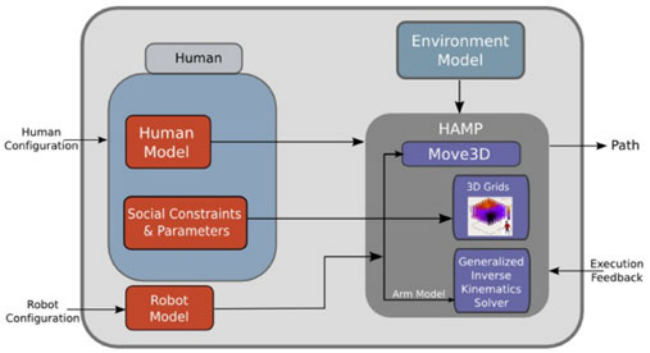


Fig. 15. Architecture of the human-aware manipulation planner module.

position data coming from GEST and the leg position data coming from HumPos modules. Having two separate data for the same human requires the fusion of the data in order to keep the human model up-to-date as correct as possible. As the laser data are more accurate than the stereo cameras, a prioritized data fusion is executed inside the planner module.

Human-aware motion planner (HAMP) maintains permanently the 3-D structures of the humans in the environment. The `STANDING/SITTING` states are also applied in this stage with a simple assumption of human height: If a person is shorter than 1.40 m, he/she is considered as `SITTING`. Although this is a simple simplification, the system can be further improved to communicate with a complete activity/state recognition system (e.g., [32]).

The actual input modules are chosen to illustrate the capabilities of the planner. Other type of sensor modalities can be used for the same planning framework (e.g., [33]).

With all these information, the planner module computes a path for the robot's arm and sends it to the execution module, Xarm. This module transforms the path to a limited jerk trajectory whose speed profiles are proved to be similar to humans' [34].

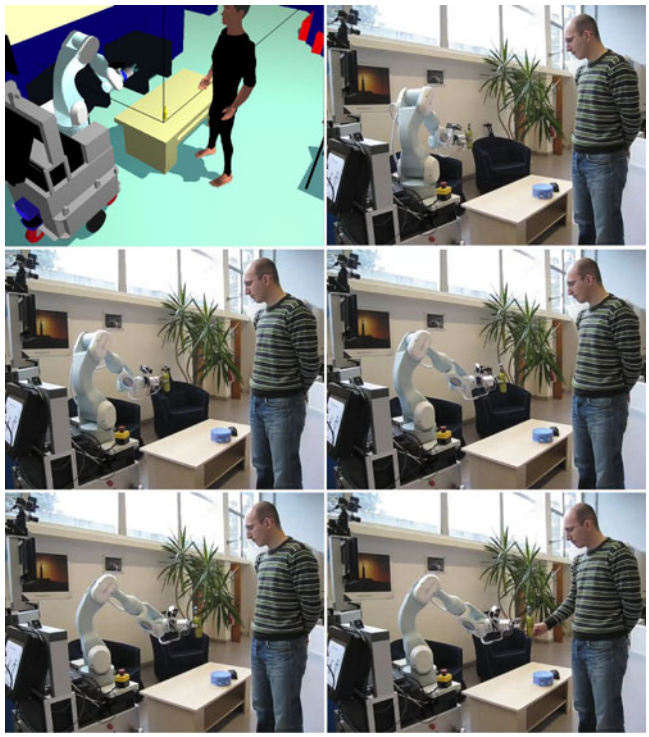


Fig. 16. Jido hands over a bottle to a standing person. The position where the object transfer will occur is chosen by the planner, and motion toward this point is performed.

In case of an important change in the human position or orientation, the system stops the execution, replans a new path, and executes this new path. The decision of the change is performed by an arbitrary value that can be tuned by the user.

B. Robot Integration Results

The motions of the robots are tested in various scenarios. Fig. 16 illustrates a scenario where the robot hands a bottle to a standing person. With the leg and head/hand detection modules, the planner updates its human model and plans a path toward a position of its choosing (the OTP) by considering that the person is standing and facing it. In this example, the person in front of the robot is considered as a right-handed person.

The planner generates and sends the path in the form of successive configurations to the execution modules. Once the robot reaches its final position, it waits for the human to pull the bottle. Once the bottle is handed, the arm reaches to its original configuration. The sequence of moving arm, detecting object pull, opening the gripper, and returning to original position is performed either by tool command language scripts passed to the robot by a human controller or by the supervisor situated in the decisional level of the LAAS architecture. We have also integrated a detection script to the supervision system in order to detect if the human is moving his hand toward the robot. If that is the case, the robot overwrites its calculated OTP and moves toward the person's hand.

Fig. 17 shows the importance of the sequencing of the actions in a handing over task. In this scenario, Jido reaches its arm to

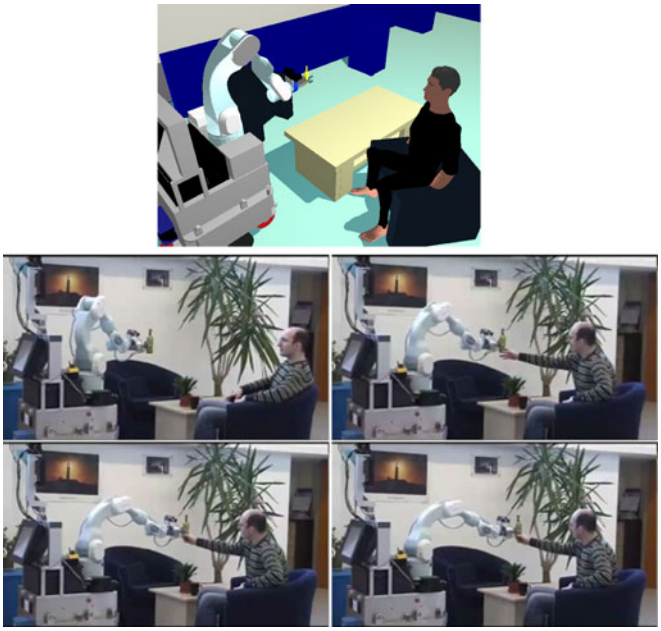


Fig. 17. When the actions of a "hand over an object" task are run sequentially, the robot only releases the object when the arm reaches its position. For a person who wants to take the object earlier, this behavior causes discomfort in the interaction.

Fig. 18. Scenario where two persons sit around the robot and ask the robot to pass a bottle from one to the other.

hand over a bottle. The person, being impatient, tries to take the bottle before robot arm reaches its destination. As the robot only releases the bottle once it reaches its destination, this behavior causes discomfort.

In order to solve this problem, a separate force detection module and the arm motion actions are run in parallel and an interrupt is sent to the execution if a pulling force is applied to the object. This parallelization allows the robot hand over the object during its motion whenever the human desires. Scenarios illustrated in Figs. 17 and 18 are performed with this parallelization.

A final example of motions is given in Fig. 18 illustrating a scenario where two persons sit around the robot. The person on the right asks the robot to take the bottle and pass it to the other person. After taking the bottle, the robot plans a path where it retracts its arm, enters the field of view of the human on the left, and then hands over the bottle.

Another functionality that is added to the planner module is the ability to bypass the OTP computation process and plan a path directly to human's hand. In addition, a monitor detecting human reaching motion is implemented in order to detect if the person performs a reaching gesture to ask for the object.

In this scenario, the planner generates a path toward the hand of the human. Once it grasps the bottle, it turns its cameras to the other side and waits for the reach gesture from the other person. Once the human asks for the object, the robot plans a path by taking into account the safety, the visibility, and the posture of the human.

In the illustrated examples, each cycle of planning is performed in $\sim$6 s on a Pentium 4 3.2-GHz computer. We believe that planning algorithms can be drastically optimized in order to allow shorter planning times. One should note that the time that the planning takes depends on the number of backtrack that the OTP, object path, and robot path computation takes. The planner did not have any backtrack in the aforementioned examples since the object position and its path were fairly unconstrained.

C. Evaluation

The overall behavior of the robot equipped by the proposed human-aware manipulation planner has been evaluated first with subjective questionnaire-based user studies in [35]. In this study, the subjects have found the robot's behavior not threatening or not causing any stress.

A physiological user study has been conducted in order to set up an objective evaluation of robot's motion behavior synthesized by our human-aware manipulation planner [36]. In this study, three different ways of handing over an object are evaluated according to galvanic skin conductance response, occulometry, and electromyogram data from 12 users. The subjects have been asked to sit on a chair placed in front of the robot (see Fig. 19). The study begun with the robot holding a bottle in its resting position with the arm folded.

We have evaluated three different motions: 1) a motion planned by the human-aware planner with moderate velocity and force detection; 2) a faster shortest path motion; and 3) a motion planned by the human-aware planner with slow velocity and without force detection.

Fig. 19. Experimentation setting.

TABLE I
PHYSIOLOGICAL SENSORS FINDINGS AND OCCULOMETRY RESULTS ACCORDING TO THE THREE MOTIONS

Sensors p-value	Motion-1	Motion-2	Motion-3
Fixations (in ms) p = .002	124.00 ($\pm$2.88)	160 ($\pm$15.03)	147 ($\pm$5.41)
Number of saccades p = .001	1 ($\pm$0.960)	0.28 ($\pm$0.61)	3 ($\pm$1.66)
GSR (in μS) p = .027	1.38 ($\pm$0.39)	3.42 ($\pm$0.78)	1.22 ($\pm$0.33)
Electromyogram (in μV) p = .009	19.60 ($\pm$3.52)	27.30 ($\pm$5.44)	32.45 ($\pm$6.95)

TABLE II
SUBJECTIVE EVALUATIONS FOR EACH MOTION

Subjective p-value	Motion-1	Motion-2	Motion-3
Legibility p $<$ .001	7.33 ($\pm$1.18)	4 ($\pm$0.72)	3.58 ($\pm$0.54)
Safety p $<$ .001	7 ($\pm$0.39)	2.25 ($\pm$ 1.05)	4.66 ($\pm$ 0.57)
Physical comfort p = .003	6.33 ($\pm$0,43)	2.83 ($\pm$0,92)	1.83 ($\pm$1,03)

Table I shows the obtained results from the physiological sensors. Motion-1, which is the human-aware motion, has appeared to be the one that requires the minimum amount of effort of human arm (electromyogram results). The fixations and the GSR values are significantly lower than Motion-2 and a little higher than Motion-3. We believe that this result comes from the velocity of the robot.

In the same experiment, we have conducted a questionnaire-based survey where we asked subject to evaluate each motion's predictability and safety along with the overall physical effort. The findings, which are illustrated in Table II, showed that motion 1 has been distinguished as more legible, safe, and comfortable than the two others. On the contrary, motion 2, the shortest path, was subjectively assessed as the most unsafe. Motion 3 was ranked as the least physically comfortable and the least legible for the subjects since its low velocity and the inhibition of the force sensor led the participants to struggle prematurely with the robot to get the bottle.

The detailed description of both studies can be found in [36].

V. CONCLUSION

In this paper, we have presented a manipulation planning framework that explicitly takes into account the presence of human by reasoning on his safety, accessibility, field of view, posture, and preferences. We have materialized this framework with human-aware manipulation planner that deals with human–robot object transfer problems.

It is very difficult to give a precise definition to the notions of acceptability, legibility, and comfort. We have proceeded with a generic cost-based approach that may be extended to different interpretations of these notions. The legibility notion is particularly difficult to evaluate since it highly depends on the user preference. This study can also be considered a first step toward integrating this fuzzy notion into the planning.

This planner is designed specially for "robot handing over an object" scenarios by decomposing the problem into three steps: 1) Finding an "acceptable" object position where the transfer will happen; 2) calculating a safe and visible path for the object as if it is a free flying body; and, finally, 3) adapting the robot body to follow the object path in a way that the robot's intentions are clearly expressed.

With this approach, the motion planner breaks the standard human-effort-based interaction and allows the robot to take initiative by letting it to compute the place where the interaction will take place. This property is also a brick to fill the gap between symbolic and geometric reasoning.

We also presented the integration of the planner into a complete robot control architecture [26] and its link with perception and execution modules. The human-aware manipulation planner has been integrated into Jido platform and evaluated by neutral subjects. According to video-based user study [35], the participants evaluated the motions of the robot as natural and comfortable without causing any unpleasantness.

VI. FUTURE WORK

This work helped us to frame questions such as the need for more elaborate models of human–robot tasks based on perspective taking and interactive execution, and the development of methodology to measure and validate HRI schemes based on physiological measures.

One of the most vital notion to ensure safety and comfort in HRI is the dynamics of the motion. In this study, we were only interested in and proposed solutions for the shape of robot motions. In real world, we cannot fully guarantee the safety if the speed of the robot remains uncontrolled. Even though we employed the classic approach of "path planning $\rightarrow$ path to trajectory transformation $\rightarrow$ trajectory execution" by using limited jerk execution module, this approach causes the loss of many solutions. In order to fully ensure safety and comfort, we have to by pass the intermediate level of path to trajectory transformation and incorporate the speed (also the acceleration and the jerk) into the planning loop.

In the described system, the combination weights of the functions are chosen arbitrarily by the developer. This possibility to tune the system can be helpful in a future application where the user can personalize the robots behavior to his/her preference.

An interesting extension to our study would be to integrate a learning mechanism, such as inverse reinforcement learning, that would find the optimal weights of combination of the described functions based on the robot's previous experience to the user.

A limitation of the human-aware manipulation planner is the necessity to have a fixed robot base. When the planner generates a path, the base of the robot is considered as not moving. This results a motion that only involves the upper body of the robot when handing over an object. In the user study that we have conducted [6], most of the subjects evaluated the motion of the arm during the motion of the base as most comfortable. Using the human-aware manipulation planner, along with the perspective placement [37] system and the human-aware navigation planner [9], allows the robot to choose a suitable robot position for manipulation task and to navigate to that point. A more integrated approach is under development and employs probabilistic roadmap approaches (more precisely T-RRT [38]) at the final stage of the algorithm (finding robot path to follow object path) by allowing the robot to move its base while following the object motion including fine orientations [39]. Even though this approach proved to be powerful and probabilistically complete, an important challenge lies on its computation speed performance to be applicable to HRI scenarios.

Although looking at the object improves the legibility of the motion, in an ideal case, the robot should try to obtain a joint attention with the human on the object. This would require a chain of motion protocols, such as looking at the object, looking at the human, looking at the object again, etc. An interesting work would be to integrate these motion protocols and their effect to the cost functions in order to generate a more natural robot behavior. The naturalness of the robot motions would also improve if the const functions evaluate not only human's posture but the whole body posture of the robot as well.

An interesting research direction is "planning for the human." For a robot helping a person, there will be moments that the robot will not find a solution satisfying both the safety and the feasibility of the task without moving the human. Therefore, the robot should also reason on how costly the human motions will be in order to plan for itself and for the human to find an optimal plan that satisfies the task.

The choice of comfortable object orientation is also an important aspect to consider. Ideally, a robot that hands over an object should make the grasp as easy as possible to the human. Toward this end, we plan to integrate a grasp planner [40]–[42] to our framework. Another improvement of the system would be to take into account the occlusion of the object caused by the robot itself. An occlusion ratio function attached to the object can be integrated to the inverse kinematics solver as a task in order to prioritize robot postures that would not hide the object. In a parallel research direction, we are also working on creating reachability and visibility maps around the human in order to better evaluate object's affordances in a particular position [43].

Finally, integrating human-aware navigation and manipulation planners to the supervisor [44] and the task planner [45] is one of the checkpoints of a complete decisional framework for human–robot interactive task achievement.

REFERENCES

[1] R. Alami, A. Albu-Schaeffer, A. Bicchi, R. Bischoff, R. Chatila, A. D. Luca, A. D. Santis, G. Giralt, J. Guiochet, G. Hirzinger, F. Ingrand, V. Lippiello, R. Mattone, D. Powell, S. Sen, B. Siciliano, G. Tonietti, and L. Villani, "Safe and dependable physical human–robot interaction in anthropic domains: State of the art and challenges," presented at the IEEE/RSJ Conf. Intell. Robot. Syst., Beijing, China, 2006.

[2] A. Bicchi and G. Tonietti, "Fast and soft arm tactics: Dealing with the safety-performance trade-off in robot arms design and control," *Robot. Autom. Mag.*, vol. 11, no. 2, pp. 22–33, 2004.

[3] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury. (2004). Playing it safe [human-friendly robots]. *IEEE Robot. Autom. Mag.* [Online]. vol. 11, no. 2, pp. 12–21. Available: http://dx.doi.org/10.1109/MRA.2004.1310938.

[4] M. Lohse, K. Rohlfing, B. Wrede, and G. Sagerer, "Try something else! when users change their discursive behavior in human–robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, 2008, pp. 3481–3486.

[5] K. Sakata, T. Takubo, K. Inoue, S. Nonaka, Y. Mae, and T. Arai, "Psychological evaluation on shape and motions of real humanoid robot," in *Proc. IEEE Int. Workshop Robot Human Interact. Commun.*, Okayama, Japan, Sep. 2004, pp. 29–34.

[6] K. L. Koay, E. A. Sisbot, D. A. Syrdal, M. L. Walters, K. Dautenhahn, and R. Alami, "Exploratory study of a robot approaching a person in the context of handing over an object," presented at the Assoc. Adv. Artif. Intell. Spring Symposia, Palo Alto, CA, Mar. 2007.

[7] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L. Nehaniv, E. A. Sisbot, R. Alami, and T. Siméon, "How may i serve you?: A robot companion approaching a seated person in a helping context," in *Proc. ACM SIGCHI/SIGART Conf. Human–Robot Interaction*, Salt Lake City, UT, Mar. 2006, pp. 172–179.

[8] E. T. Hall, *The Hidden Dimension*. Garden City, NY: Doubleday, 1966.

[9] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 874–883, Oct. 2007.

[10] D. Kulic and E. Croft. (2007, Feb.). Pre-collision safety strategies for human–robot interaction. *Auton. Robots* [Online]. vol. 22, no, 2, pp. 149–164. Available: http://dx.doi.org/10.1007/s10514-006-9009-4

[11] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *Int. J. Robot. Res.*, vol. 22, no. 5, pp. 281–297, 2003.

[12] M. Nokata, K. Ikuta, and H. Ishii, "Safety-optimizing method of human-care robot design and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, May 2002, vol. 2, pp. 1991–1996.

[13] R. Alami, R. Chatila, A. Clodic, S. Fleury, M. Herrb, V. Montreuil, and E. A. Sisbot, "Towards human-aware cognitive robots," presented at the 5th Int. Cognit. Robot. Workshop, Boston, MA, 2006.

[14] E. A. Sisbot, L. F. M. Urias, and R. Alami, "Spatial reasoning for human–robot interaction," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, San Diego, CA, 2007, pp. 2281–2287.

[15] E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan, "Supervision and motion planning for a mobile manipulator interacting with humans," presented at the ACM/IEEE Human–Robot Interaction, Amsterdam, The Netherlands, 2008.

[16] R. T. Marler, J. Yang, J. S. Arora, and K. Abdel-Malek, "Study of bi-criterion upper body posture prediction using Pareto optimal sets," presented at the IASTED Int. Conf. Model., Simul. Optim., Oranjestad, Aruba, Aug. 2005.

[17] K. Abdel-Malek, Z. Mi, J. Yang, and K. Nebel, "Optimization-based layout design," *Appl. Bion. Biomech.*, vol. 2, no. 3–4, pp. 187–196, Jan. 2005.

[18] M. Katayama and H. Hasuura, "Optimization principle determines human arm postures and 'comfort,'" in *Proc. SICE Annu. Conf.*, vol. 1, pp. 1000–1005, Aug. 2003.

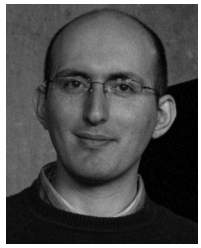
[19] M. Kolsch, A. C. Beall, and M. Turk, "The postural comfort zone for reaching gestures," in *Proc. Human Factors Ergon. Soc. Annu. Meet.*, vol. 47, pp. 787–791, 2003.

[20] T. Kang, S. Tillery, and J. He, "Determining natural arm configuration along reaching trajectory," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Cancun, Mexico, Sep., vol. 22003, pp. 1444–1447.

[21] D. Tolani, A. Goswami, and N. I. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graph. Models Image Process.*, vol. 62, no. 5, pp. 353–388, 2000.

[22] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1990.

[23] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *Vis. Comput.: Int. J. Comput. Graph.*, vol. 20, no. 6, pp. 402–417, 2004.
[24] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 352–360, Jul./Sep. 2003.
[25] T. Siméon, J.-P. Laumond, and F. Lamiraux, "Move3d: A generic platform for path planning," in *Proc. IEEE Int. Symp. Assem. Task Planning*, Fukuoka, Japan, May 2001, pp. 25–30.
[26] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *Int. J. Robot. Res.*, vol. 17, pp. 315–337, 1998.
[27] S. Fleury, M. Herrb, and R. Chatila. (1997, Sep.). "Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Grenoble, France, pp. 842–848 [Online]. Available: citeseer.ist.psu.edu/fleury97genom.html
[28] B. Burger, I. Ferrané, and F. Lerasle, "Multimodal interaction abilities for a robot companion," in *Proc. Int. Conf. Comput. Vis. Syst.*, Santorini, Greece, 2008, pp. 549–558.
[29] M. Fontmarty, T. Germa, B. Burger, L.-F. Marin, and S. Knoop, "Implementation of human perception algorithms on a mobile robot," presented at the IFAC Symp. Intell. Auton. Veh., Toulouse, France, Sep. 2007.
[30] I. Herrera-Aguilar, "Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service," Ph.D. dissertation, Dept. LAAS-CNRS, Univ. Toulouse, Toulouse, France, Sep. 2007.
[31] X. Broquère, D. Sidobre, and I. Herrera-Aguilar, "Soft motion trajectory planner for service manipulator robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Toulouse, France, 2008, pp. 2808–2813.
[32] M. Losch, S. Schmidt-Rohr, S. Knoop, S. Vacek, and R. Dillmann, "Feature set selection and optimal classifier for human activity recognition," in *Proc. IEEE Int. Workshop Robot Human Interact. Commun.*, Jeju, Korea, Aug. 2007, pp. 1022–1027.
[33] S. Knoop, S. Vacek, and R. Dillmann, "Fusion of 2D and 3D sensor data for articulated body tracking," *Robot. Auton. Syst.*, vol. 57, no. 3, pp. 321–329, 2009.
[34] M. M. Rahman, R. Ikeura, and K. Mizutani, "Investigating the impedance characteristic of human arm for development of robots to co-operate with human operators," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 1999, vol. 2, pp. 676–681.
[35] Cogniron. (2008). Cogniron ra7 deliverable, Tech. Rep. 08868. [Online]. http://www.cogniron.org/review4-open/page7/page7.html
[36] F. Dehais, E. A. Sisbot, M. Causse, and R. Alami, "Physiological and subjective evaluation of a human–robot object hand-over task," *Appl. Ergon.*, vol. 42, no. 6, pp. 785–791, 2011.
[37] E. A. S. Luis Marin and R. Alami, "Geometric tools for perspective taking for human–robot interaction," presented at the Mexican Int. Conf. Artif. Intell., Mexico City, Mexico, Oct. 2008.
[38] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based RRT for path planning in continuous cost spaces," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Nice, France, 2008, pp. 2145–2150.
[39] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortes, R. Alami, and T. Simeon, "Planning human-aware motions using a sampling-based costmap planner," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5012–5017.
[40] Y. Li, J.-P. Saut, J. Cortes, T. Simeon, and D. Sidobre, "Finding enveloping grasps by matching continuous surfaces," in *Proc. IEEE/RSJ Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 2825–2830.
[41] K. Nagata, T. Miyasaka, D. N. Nenchev, N. Yamanobe, K. Maruyama, S. Kawabata, and Y. Kawai, "Picking up an indicated object in a complex environment," in *Proc. Int. Conf. Intell. Robot. Syst.*, Oct. 2010, pp. 2109–2116.
[42] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Proc. Int. Conf. Intell. Robot. Syst.*, 2010, pp. 1579–1585.
[43] A. Pandey and R. Alami, "A perceptual level decisional framework for cooperative and competitive human–robot interaction," in *Proc. Int. Conf. Intell. Robot. Syst.*, 2010, pp. 5842–5848.
[44] A. Clodic, M. Ransan, R. Alami, and V. Montreuil, "A management of mutual belief for human–robot interaction," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Montreal, QC, Canada, Oct. 2007, pp. 1551–1556.
[45] V. Montreuil, A. Clodic, M. Ransan, and R. Alami, "Planning human centered robot activities," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Montreal, QC, Canada, Oct. 2007, pp. 2618–2623.



Emrah Akin Sisbot (M'12) received the Ph.D. degree in robotics from Paul Sabatier University, Toulouse, France, in 2008.

He was a Postdoctoral Research Fellow with LAAS-CNRS, Toulouse, France, for two years and with the University of Washington, Seattle, for one year. He is currently a Research Scientist with Toyota InfoTechnology Center, Mountain View, CA. His current research interests include human–robot interaction, spatial reasoning, navigation and manipulation planning, perspective taking, and multirobot cooperation.

Rachid Alami received the Docteur-Ingénieur degree in computer science from the Institut National Polytechnique de Toulouse, Toulouse, France, in 1983.

He joined the Robotics and Artificial Intelligence Group, LAAS-CNRS, Toulouse, in 1980, where he is currently a Senior Research Scientist and the Head of the Robotics and Interactions Group. His current research interests include the study of control architectures for autonomous robots, task and motion planning, multirobot cooperation, personal robots, and human–robot interaction.