Online ISSN 1848-3380 , Print ISSN 0005-1144
ATKAFF 52(3), 244–255(2011)

Enrico Di Lello, Alessandro Saffiotti

The PEIS Table: An Autonomous Robotic Table for Domestic Environments

UDK 004.89.031.42; 004.94
IFAC 2.8; 5.9.3

Original scientific paper

There are two main trends in the area of home and service robotics. The classical one aims at the development of a single skilled servant robot, able to perform complex tasks in a passive environment. The second, more recent trend aims at the achievement of complex tasks through the cooperation of a network of simpler robotic devices pervasively embedded in the domestic environment. This paper contributes to the latter trend by describing the PEIS Table, an autonomous robotic table that can be embedded in a smart environment. The robotic table can operate alone, performing simple point-to-point navigation, or it can collaborate with other devices in the environment to perform more complex tasks. Collaboration follows the PEIS Ecology model. The hardware and software design of the PEIS Table are guided by a set of requirements for robotic domestic furniture that differ, to some extent, from the requirements usually considered for service robots.

Key words: Service Robotics, Autonomous robotic table, Robots Ecology

PEIS stol: autonomni robotski stol za kućanstva. U uslužnoj robotici i robotici za kućanstva postoje dva glavna trenda. Klasičan pristup teži razvoju jednog složenog uslužnog robota koji je sposoban izvršavati složene zadatke u pasivnom okruženju. Dok drugi, nešto noviji pristup, teži rješavanju složenih zadataka kroz suradnju umreženih nešto jednostavnijih robota prožetih kroz cijelo kućanstvo. Ovaj članak svoj doprinos daje drugom pristupu opisujući PEIS stol, autonomni robotski stol koji se može postaviti u inteligentnom okruženju. Robotski stol može djelovati samostalno, navigirajući od točke do točke ili može surađivati s ostalim uređajima u okruženju radi izvršavanja složenijih zadataka. Ta suradnja prati PEIS ekološki model. Dizajn sklopovlja i programske podrške PEIS stola prati zahtjeve za robotsko pokućstvo koji se donekle razlikuju od zahtjeva koji se inače postavljaju za uslužne robote.

Ključne riječi: uslužna robotika, autonomni robotski stol, robotska ekologija

1 INTRODUCTION

> *Leonardo is 72 and he has a broken leg. Soon after he moves from the coach to his bed, the low table moves away from the coach, enters the bedroom and docks on the side of the bed, carrying Leonardo's book, glasses and mobile phone on its top. While moving, the table keeps track of its position with the help of the security cameras on the ceiling. Later Leonardo asks the table to bring some water. The table navigates into the kitchen, asks the fridge to open its door and to use its gripper to put a bottle of water on the table top, and returns to the bed-side.*

This vignette illustrates some of the uses of an imaginary autonomous robotic table included in a domestic environment. It also illustrates some of the capabilities of the PEIS-Table, the real autonomous table which we describe in this article.

Our vignette reflects the widespread expectation that robots will soon become part of our homes and contribute to improve the quality of our life, especially for those in need of special care like senior citizens. The most common vision in the robotics community which underlies this expectation, however, is rather different from the one depicted in our vignette. In this vision, the place of robots in domestic environments will be taken by multi-purpose, extremely skilled, often anthropomorphic robots, performing tasks that somehow resemble what a human butler would do [1–3].

An alternative vision has recently emerged, in which a multitude of *robotic devices* are pervasively integrated in a smart environment and are able to exchange information and coordinate action among them [4, 5]. In this vision, the performance of complex tasks is not achieved through the development of a very advanced robot, but through the cooperation of many simpler, specialized robotic devices distributed in the environment. The vignette above is an illustration of this "ecological" vision: the moving table, the robotic fridge and the security cameras cooperate to provide the needed robotic services. Concrete realizations of this vision are now burgeoning, and include Artificial Ecosystems [6], Ambient Ecologies [7], the Ubiquitous Robotic Space project [8], the U-RT project [9], and the PEIS-Ecology project (Ecology of Physically Embedded Intelligent Systems) [10]. We generically refer to systems of this type as “ecology of robots”.

In this paper we push the above vision further, and we propose to extend robot ecologies to also include *robotic furniture*. We believe that embedding robotic technologies inside everyday furniture, rather than inside traditional robots, may provide a smother path to bring robotic services into home environments. Robotic furniture may have advantages over more traditional robotic devices in terms of acceptability, cost-effectiveness, and modularity. We make our proposal concrete by describing the realization of an autonomous table like the one in our vignette.

Robotic furniture has never been incorporated in a robotic ecosystem until now, although this possibility has been suggested before [11, 12]. A few examples of robotic furniture have been reported in the fields of Ambient Intelligence [13, 14] and Interaction Design [15, 16], but these are usually stand-alone objects that are manually operated. By contrast, the robotic table proposed in this paper has autonomous navigation capabilities, and it is fully integrated in a smart environment.

In realizing a piece of autonomous robotic furniture for use in domestic environments, a number of requirements should be taken into account [17, 18]. Interestingly, some of these requirements induce some constraints to the autonomous navigation problem that are different from the ones usually considered in the literature on mobile and service robotics, and call for different approaches. In this paper, we discuss the requirements for a domestic robotic table, and we present a concrete realization of such a table which satisfies these requirements. Our table is set in the context of the PEIS-Ecology project [10], and we therefore call it the PEIS-Table.

2 REQUIREMENTS FOR A DOMESTIC MOVING TABLE

Our main assumption in designing robotic artifacts for domestic environments is that these artifacts should not be perceived as foreign bodies by the users, but rather as a natural extension of their usual, familiar environment. We believe that this is an important factor to ensure the acceptability of domestic robotic technology, especially in the case of senior users. A similar assumption is usually made for other types of technologies [19], including assistive technology [20]. Previous studies in robotics for elderly have also shown that the acceptability of a robotic service is strongly influenced by the physical appearance and motion behavior of the robot [21, 22]. The requirements that follow are inspired by the above assumption, and they have guided the design of our PEIS-Table. We expect that similar requirements should apply to the design of any piece of robotic furniture.

2.1 Hardware Requirements

The first general requirement for the design of the PEIS-Table hardware is that it should be *familiar*, that is, it should look as much as possible like a regular table. Users should feel it natural to find the table around them and should not be threatened by its presence. User should also perceive the table as familiar from a functional point of view: they should feel it natural to place objects on it, or to move it around.

The above means that the mechanical and electronic parts should not be visible to the user, or they should be concealed as decorative elements. This also applies to the wheels, whose placement should make them barely visible while sticking to a kinematic model that allows high maneuverability in reduced space. As for the material and colors used, we did not pose any special restriction although in our design we opted for a wooden appearance, in style with the target environment.

The second general requirement is that the device should be *non-invasive*, that is, it should have minimal impact on the existing environment beyond the fact that it provides new functionalities. In particular, potentially dangerous active sensors should be avoided, and noise emissions should be low. Energy consumption, weight, construction cost, and maintenance cost are other parameters that should be kept as low as possible to minimize impact. These factors suggest for instance that laser scanners, which are largely used in today’s mobile robots, should be avoided here.

In a robot ecology context, there is a third requirement: the device should be *ecology aware*, that is, it should be ready to interact with the other devices in the environment whenever it is embedded in a smart environment, as illustrated by the Leonardo’s scenario. This implies that the table should be equipped with the necessary communication hardware. It also implies that the table does not need to be overloaded with sensors and actuators. In a robot ecology perspective, the table only needs enough sensing and actuation capabilities to perform the minimum set of tasks that it is meant to perform — in our case, point-to-point navigation and docking. More complex tasks will be performed with the help of other sensors and actuators in the environment, like in the Leonardo scenario.

2.2 Software Requirements

The PEIS-Table is meant to operate in a domestic environment, which is typically mildly dynamic and populated by human beings. This induces a number of requirements on its navigation and control software.

First, the motion of the table should be *perceived as safe*, meaning that it should be both actually safe and perceived as such by the humans in the environment. For instance, the table should not get too close to objects and walls, movements should be smooth and not too fast, and there should be few and predictable key turning points. Predictability in particular is a key factor for avoiding collisions in human-human interaction on roads [23], and is an important precondition for trustability. Accordingly, the paths planned by the robot should not aim to minimize length or time, but to maximize clearance from obstacles and predictability. Moreover, the motion should be smooth and safe even in the presence of unknown obstacles and uncertainty in the sensor data. In the case of a robot butler, other constraints on the acceptable paths could be posed based on the visibility of the robot from the human point of view [24]. These, however, were considered less important for a moving table due to its intrinsically less threatening nature.

The second requirement, partly a consequence of the first one, is that the table should *maintain a map* of the environment and perform global localization on it, in order to plan and follow safe routes between positions in the home. The map should include occupancy information, but additional information may also be useful, e.g., names of places to be used for human-robot interaction. Providing a rough *a-priori* map is acceptable, provided that the table is able to dynamically update this map to account for displaced or unmapped obstacles. However, using a full-fledged dynamic SLAM algorithm (e.g., [25,26]) would be undesirable in our case because of the needed training period, because of the high computational requirements, and because most current methods to build occupancy maps assume expensive sensors like laser scanners or stereo cameras. In the case of the PEIS-Table, we have opted for a simple method based on fuzzy occupancy grids built from sonar data, coupled with a commercial indoor GPS.

Finally, the requirement for the table to be *ecology aware* has also a software side. The table should be able to recognize and participate in a robot ecology if it is placed in one: this means that it should include suitable communication and cooperation software to be integrated in the specific robot ecology. In the case of the PEIS-Table, this is realized by building the navigation software on the top of the PEIS-Ecology middleware [10].

3 THE PEIS-TABLE: HARDWARE

The PEIS-Table has been built starting from a commercial table (LACK, from IKEA) and a commercial robot base: an ActivMedia AmigoBot, augmented with a PC-104 board running Linux, a Ni-Mh high capacity battery and a IEEE 802.11 bridge. In order to obtain the desired table-like appearance, the original parts were disassembled and placed in a custom-made aluminum structure. This structure was used as the new base for the LACK board, and enclosed by wooden panels. A ring of LEDs was placed near the bottom, that indicate the table's status by their color. The table has standard height of 45 cm. Figure 1 shows the assembled PEIS-Table, while Table 1 summarizes its specifications.

Table 1. PEIS-Table hardware and specification

Mechanics	2 Motors with 500 ticks encoders
Battery	12V, 13 AmpH NiMh
Sonar range finders	8 + 4
PC-board	EPIA 900 with VIA CPU, 256 MB RAM
Communication device	IEEE 802.11 bridge
Localization device	ETRI StarLITE
Max speed	300 mm/s
Tested payload	2 Kg

In order to obtain a good-looking proportion between the base and the board, the new chassis had to be narrower than the original one. The two driving wheels have been placed on the sides of the base, and four spherical metal casters have been added under each corner to improve stability. A custom-made transmission with a 1:3 ratio was mounted downstream the original one in order increase the motor torque and hence the table maximum payload. An omni-directional driving mechanism was also considered, but rejected because of the added complexity.

As for the sensors, we decided to avoid the use of laser range finders because of the non-invasiveness requirements discussed above, including safety, energy, cost and weight considerations. Instead, we have opted for sonar sensors, which provide a reasonable tradeoff between invaseveness and reliability. The sensory system is composed of 2 arrays of sensors. The first array is composed of eight Polaroid transducers, providing reliable measure up to 2.5 m, suitable as range sensors for map-building and obstacle avoidance in a small environment. This array is hidden under the table board, and distributed to span an angle of $180°$ in the front side of the table.

The second array is composed of four separated emitter-receiver sonars, able to measure distances down to 25 mm, inset into the edges of table board. Compared to the first array, the second array features better precision on small distances and reduced blind-distance. This array is used to provide accurate distance measurements during the dock-

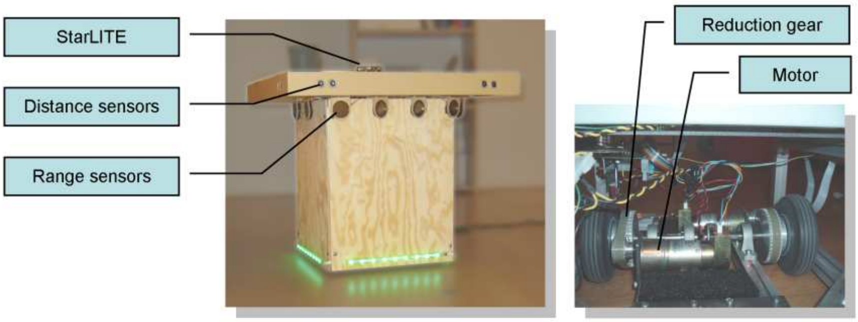

Fig. 1. The PEIS*-Table hardware, showing the main sensors and actuators.*

ing procedure, when small corrective maneuvers are performed according to the front and lateral distances from the docking reference surfaces.

The motors are equipped with encoders, but the odometric precision of the PEIS-Table is inherently hindered by several factors, including: the non-linearities introduced by the custom made transmission; the variable friction in the caster wheels due to dust infiltration; and the unpredictable nature of wheel/floor interaction in a domestic environment. Because of this, and because of the difficulty to obtain precise and reliable self-localization from sonar data, we decided to equip the PEIS-Table with an indoor GPS for global localization.

The system of our choice has been the StarLITE, developed by ETRI [27].[1] The StarLITE system consists of a IR camera sensor placed on the robot, and a set of infrared light emitting tags placed in the ceiling at known position. The sensor is very small, which helped to keep the table-like appearance of the PEIS-Table. The sensor provides an estimate of its position and orientation in a global reference frame at 30 Hz, with a typical precision of about 4 cm in position and 1° in orientation. While the use of an indoor GPS system has a negative impact on the installation and maintenance cost of the PEIS-Table, these drawbacks are mitigated by the increasing diffusion and decreasing cost of these systems.

4 THE PEIS-TABLE: SOFTWARE

The control architecture developed for the PEIS-Table is shown in Figure 2. This architecture has been implemented in C, and interfaced with the robot hardware using Player [28]. Many of the PEIS-Table functionalities have been realized using a fuzzy-logic approach, which proved to be effective in coping with the different sources of uncertainty which characterize our domain. The next subsections describe the main functionalities appearing in Figure 2.

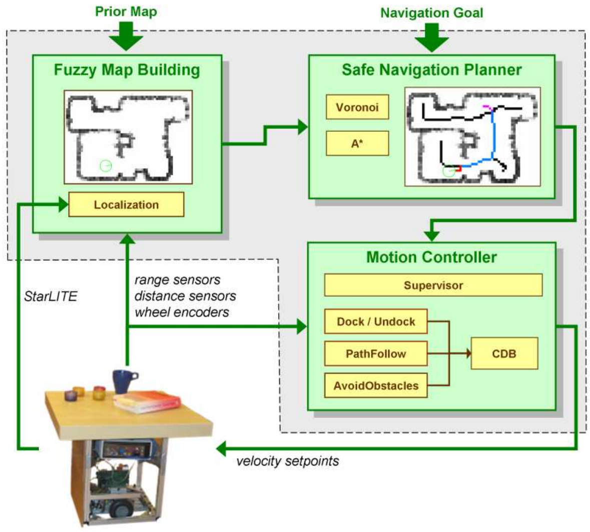

Fig. 2. Schematic control architecture of the PEIS*-Table.*

4.1 Fuzzy Map Building

The PEIS-Table maintains a geometric map of the environment in the form of a global occupancy grid. The map is initialized from prior knowledge, if available, and updated during navigation using data from the first sonar array, together with global position information obtained through a combination of the StarLITE system and odometry.

The choice of the approach to map building was influenced by the requirements for the PEIS-Table, which in turn led us to use sonars as our main ranging sensors. Unfortunately, the use of sonars in a small and cluttered environment like the domestic one amplifies the well known shortcomings of this kind of sensors. Phenomena like sonar beam multiple reflection or missing echoes, together

[1] The StarLITE system used in the PEIS-Table is a prototype kindly provided by Dr. Wonpil Yu, ETRI, Korea.

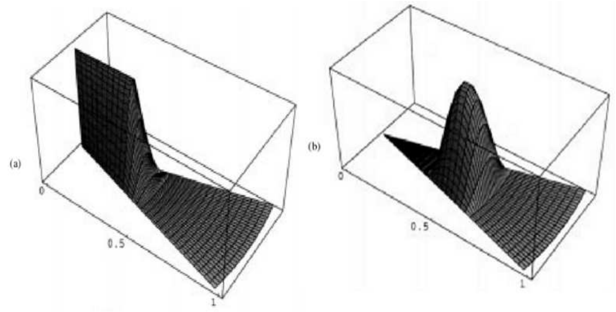

*Fig. 3. **Empty** (a) and **Occupied** (b) fuzzy models for sonar reading interpretation. (From [29].)*

with the intrinsic dynamic nature of the environment, call for a map building approach able to handle misleading measures in a robust way. We have opted for the approach based on fuzzy grid-maps proposed by Oriolo and colleagues [29], since this approach has been shown to cope well with the noisy sonar data while keeping the computation simple.

Like any approach based on occupancy grids [30], the key idea is to discretize the environment in evenly-spaced cells, and to estimate the presence of an obstacle in the region corresponding to each cell. Fuzzy grid-maps differ from conventional occupancy grids in that uncertainty about the occupancy status of each cell is represented using fuzzy logic rather than probability theory. Fuzzy logic is used at two levels: in modeling noisy sonar readings (fuzzy sensor models), and in incorporating the new information in the current map (fuzzy sensor fusion). An important consequence of using fuzzy logic is that the degree of certainty in one cell being occupied is decoupled from the degree of certainty in that cell being free: in particular, a cell for which we do not have any information has both zero certainty of being occupied and zero certainty of being free.

More specifically, noisy data from the sonar sensors are using a pair of fuzzy models, **Empty** and **Occupied**, represented by two fuzzy membership functions, μ_e and μ_o. Given a single range reading r_i^k, where r is the range and i, k are the sensor and time index, and an arbitrary cell c, the value of $\mu_e(c, r_i^k) \in [0, 1]$ gives the degree of certainty that the space covered by c is empty, while $\mu_o(c, r_i^k)$ gives the degree of certainty that this space is occupied by an obstacle. Figure 3 gives a visualization of these two fuzzy models. Intuitively, these models encode the following knowledge. Given the single reading r_i^k, there is evidence that there is an obstacle somewhere along an arc of radius r_i^k and width equal to the sensor's field of view (here, about $25°$); hence, points around that arc have high value of μ_o. There is also evidence that there is no occluding obstacle in between; hence points closer than r_i^k have high value of μ_e. Points beyond r_i^k would be occluded, hence both μ_e and μ_o are zero, indicating total lack of evidence about those points. Oriolo and colleagues [29] have shown that this approach allows a realistic modeling of the uncertain data provided by sonar readings, and that the resulting fuzzy grid-maps are robust in managing this uncertainty, while being computationally efficient.

Fig. 4. Fuzzy occupancy grid-maps representing the empty (left) and occupied (right) space. Darker cells have higher degrees of certainty of being empty (left) or occupied (right). The green circle indicates the current robot position.

Since information about occupancy and about emptiness is kept separate, the method by Oriolo and colleagues builds two separate fuzzy grids, respectively representing the empty and the occupied space. Figure 4 shows two such grids built in our test environment. Some cells do not belong to either map: these are cells for which no information can be inferred from the available sonar readings, e.g., cells behind a wall.

In a domestic environment, the position of walls and heavy furniture can be considered constant over time. Hence, a partial *a-priori* occupancy map of the space can be pre-loaded into the map building module, thus allowing the robot to generate topologically correct paths even before the environment has been fully explored. Fuzzy occupancy grids give us some degree of freedom on the way to include prior information, depending on the meaning of this information. In our application, prior information about walls and heavy furniture is used to set the corresponding cells in the *occupied* map to 1, meaning that those cells are known to be occupied with certainty and are not expected to change. Prior information about empty parts of the space, however, are only used to set the corresponding cells in the *empty* map to a small value $\beta < 1$: this value induces a bias in the planning process, but it is easily overcome when data are received from the sonars that indicate the presence of an obstacle. In our experiments, we have set $\beta = 0.5$.

The map building module also provides a self-localization functionality, which is based on the StarLITE system as discussed above. Temporary failures of this system, e.g., due to incomplete coverage or loss of radio connection with the IR tags, are compensated by integrating odometric data into the StarLITE system.

4.2 Safe Navigation Planning

When a navigation task is requested, the path planning module exploits the information in the occupancy maps to compute a path to the goal position. As discussed above, the main requirement for this path is to generate motions that are both safe and perceived to be safe by the humans who share the environment with the PEIS-Table.

Many approaches to solve the robot path planning problem have been proposed in the literature, including both deterministic and probabilistic methods. Most approaches represent the connectivity of the environment by a set of nodes and arcs, and generate an optimal path between the start and the goal position using graph search algorithms, where optimality is typically related to distance. These approaches often result in paths that run close to the walls or furniture, which would not be perceived as safe by a human. Moreover, these paths may be hard to follow smoothly because of the conflicting controls that may be generated by the path following and the collision avoidance modules in the proximity of obstacles.

In order to produce paths that fulfill the perceived safety requirement, we restrict the navigation of the PEIS-Table, whenever possible, to the locus of the points which maximize the distance from the two closer obstacles. This corresponds to the *Generalized Voronoi Diagram* (GVD) extracted from the occupancy grid of the environment. To go from a start position x_s to a goal position x_g, then, our path planner generates a path consisting of three legs: (1) an initial leg that goes from x_s to x_1, the closest point to x_s that belongs to the GVD; (2) a path within the GVD from x_1 to x_2, the closest point to x_g that belongs to the GVD; (3) a final leg that goes from x_2 to x_g. An example of such a path is shown in Figure 5: the short initial and final legs are drawn in red, while the main GVD leg is drawn in blue.

The computation of the GVD is performed any time the map is updated: in our system, this is done at 5 Hz in order to provide prompt reactivity to newly observed obstacles and to moving obstacles, e.g., people. In practice, the computation of the GVD may be expensive, and we compute an approximation of it by exploiting its similarity with an image processing transformation named *skeletonization*. The skeleton of a generic closed curve is defined as the locus of the centers of all maximal inscribed hyper-spheres. The computation of the skeleton is done using standard image processing techniques as follows. First, the values contained in the *empty* fuzzy occupancy grid are thresholded to produce a binary grid; second, a morphological dilation is performed to "grow" the obstacles in the map by a given safety radius, followed by a morphological closure; finally,

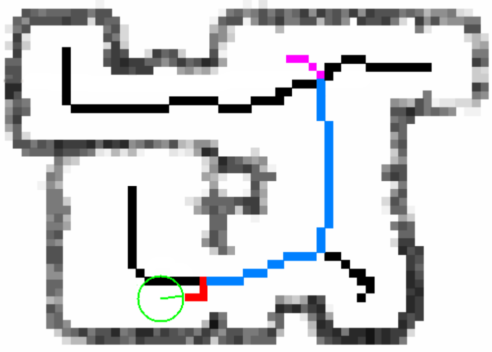

Fig. 5. An example of path planning using the Generalized Voronoi Diagram.

a thinning algorithm [31] is iteratively applied to compute the skeleton of the filtered grid. The skeleton computation is done in less than 1 msec in the PEIS-Table.

The full path planning from an initial cell c_s to a goal cell c_g, then, is performed as follows:

1. An A* algorithm is run on the thresholded *empty* grid, with c_s as start and the cells in the skeleton as goals; this results in a path in free space that connects c_s to the closest cell on the skeleton; this cell is named c_1.

2. In a similar way, a path is computed that connects c_g to the closest cell on the skeleton; this cell is named c_2.

3. A* is run on the skeleton grid, with c_1 as start and c_2 as goal; this results in a path on the skeleton that connects those two cells.

4. The three paths are connected together, and they are converted in a list of (x, y) way-points that are passed to the path following controller.

The entire procedure involves three calls to A* on three subspaces of the *empty* grid. Its complexity is therefore at most three times the complexity of A* on the same grid, although in typical cases the number of cells involved is much lower than the cells in *empty*. In our case, path planning is performed in a few milliseconds. This allows us to recompute the path frequently during the navigation (at 1 Hz in our implementation) in order to provide prompt reactivity to changes in the map.

When the start and goal points are close, a direct path may be more convenient than one that passes through the skeleton. For this reason, we also compute (again through A*) the direct path between the start and goal point. When this path is shorter than the distance to cover for the first two steps of the skeleton path-planning, then the direct path is chosen.

4.3 Motion Control

Motion control is implemented by a set of *fuzzy behaviors*, that are coordinated by a fuzzy supervisor module. Each fuzzy behavior implements a simple control policy for a given objective – e.g., follow a given path, dock to a given pose, or avoid obstacles. This policy is coded using fuzzy rules, which associate classes of sensor readings to control actions. See [32] for an overview of the use of fuzzy logic to realize behavior-based robot control.

Different behaviors can use inputs from different sensors, but they have in common the control outputs, which are a translational and rotational velocity. The output from different behaviors are combined by the fuzzy supervisor module, according to the current context. For instance, if a dangerous situation is detected, the relative weight of the output from the obstacle avoidance behavior is increased. This way to combine behavior is called context-dependent blending [33].

The four main behaviors implemented in the PEIS-Table are: FollowPath, AvoidObstacles, Dock, and Undock.

The FollowPath behavior takes as input a sequence of way-points generated by the path planner plus the current position estimate produced by the StarLITE system. The control outputs are set by the fuzzy rules depending on distance and bearing of the next way-point relative to the PEIS-Table.

The AvoidObstacles behavior is constantly active during the execution of every task, albeit with a degree of activation that depends on the current situation, except during the final docking step (see below). This is a purely reactive behavior, in that its output only depends on the current sonar readings.

The Dock and Undock behaviors are the most complex behaviors in the PEIS-Table. These can be considered *hybrid* behaviors, since they include a discrete part, implemented through finite state automata, and a continuous part, implemented through sets of fuzzy rules. Moreover, during execution these behaviors switch from a goal oriented strategy to a purely reactive one.

The Dock behavior is designed to drive the PEIS-Table to a *docking position*: this is specified by a (x, y) approach position, a required heading θ, and up to four side and front goal distances. The behavior can cope with different shapes of the docking area, by specifying different numbers of goal distances: 4 for a box area, 3 for a corner area, 2 for a wall area. Intuitively, docking with more constraints will result in a more predictable final position of the table.

Posture control of non-holonomic mobile robot is known to be a difficult problem. In order to perform such a task, the docking procedure was broken down in three steps, each one performed using a different set of fuzzy rules:

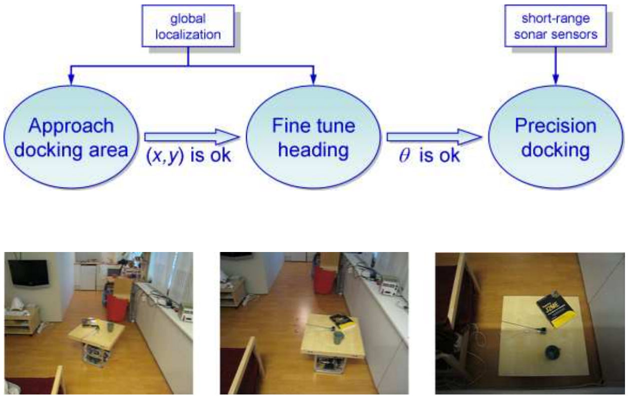

Fig. 6. The three steps of the docking behavior.

- **Step 1** The path planner is invoked and the approach position is reached using the same rules as in the *FollowPath* behavior;
- **Step 2** When the approach position is reached within a given error, a different set of rules is activated, to perform fine tuning of PEIS-Table heading;
- **Step 3** When the required heading is reached, a purely reactive set of rules is activated that perform corrective maneuvers relying on data form the proximity sonars to reach the required lateral distances. The *AvoidObstacles* behavior is de-activated in this step.

Note that in the first two steps the only input used is the table's pose produced by the localization system, while in the third step the only inputs are the distance measurements provided by the proximity sonars. Figure 6 illustrates the three steps of the docking behavior.

The following is an excerpt of the fuzzy rules used during the third step:

IF CLdist_big $\wedge$ CRdist_big
    THEN (Forward,20)
IF Ldist_big $\wedge$ $\neg$(CRdist_ok $\wedge$ CLdist_ok)
    THEN (TurnLeft,15)
IF Ldist_small $\wedge$ $\neg$(CRdist_ok $\wedge$ CLdist_ok)
    THEN (TurnRight,15)
IF CRdist_ok $\wedge$ CLdist_ok
    THEN (Forward,0)
IF Ldist_ok
    THEN (Forward,0)
IF CRbiggerCL $\wedge$ Ldist_ok
    THEN (TurnLeft,15)
IF CLbiggerCR $\wedge$ Ldist_ok
    THEN (TurnRight,15)

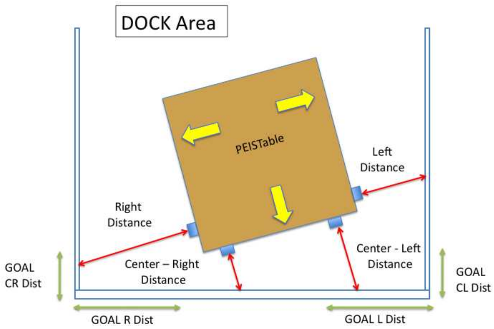

Fig. 7. Illustration of the relevant quantities used in the fuzzy rules for precision docking.

The meaning of the fuzzy predicates in the rule antecedents is illustrated in Fig. 7. These predicates state that the measure of a lateral distance is bigger, smaller or comparable to the goal one, respectively. When a distance measure coming from each short range sonar is received, the truth value of each fuzzy predicate is evaluated. The truth value of the antecedent of each rule is computed from the truth values of its constituent predicates using the operators of fuzzy logic. The consequents of all rules denote parameterized fuzzy control values. These consequents are weighted by the truth values of the corresponding antecedents and combined into an overall fuzzy set of control values. A crisp control value is finally extracted from this combined fuzzy set through defuzzification, and this value is sent to the actuators.

The Dock behavior performed well in our navigation experiments. Although no systematic validation was made of the behavior alone, we have observed that the table docked with a maximum error of $\pm 10\,cm$ and $\pm 15°$ (approximately) in 9 out of 10 runs, all of which involved docking at a corner. In one run, the table failed to reach the approach position due to a large localization error. This performance was acceptable for the demonstration goals of this table.

The Undock behavior performs similar steps as Dock but in the reverse order. The fuzzy rules used in the other behaviors have a similar format as the one above, and are computed in a similar way. See, e.g., [33] for more details on the use of fuzzy rule-based behaviors in robot control

5 DEPLOYMENT IN A PEIS ECOLOGY

The PEIS-Table has been incorporated in our PEIS-Home facility and extensively tested both in isolation and in cooperation with other devices in the environment.

5.1 Simulations

Before testing the overall structure of the PEIS-Table in a real environment, several simulations were ran using

Fig. 8. Two views of the PEIS-Table operating in the PEIS-Home. Top: navigating inside the living room. Bottom: docked at the fridge.

Player, a network server for robot control which provides a simple interface to the robot's sensor and actuator, and Stage, its simulation front-end [28]. The PEIS-Table and the domestic environment used for the real test have been modeled in Stage, and a partial map representing the occupancy of the environment was pre-loaded in the map-building module.

The navigation and docking behaviors were tested and tuned in simulation in a large number of situations including different obstacles, making sure in particular that the docking maneuvers could be successfully completed from different positions of the modeled environment. These tests were intended as a preliminary step before the actual validation of the physical table in the real environment.

5.2 Validation in the PEIS Home

The PEIS-Home is an experimental environment built to test the concept of PEIS-Ecology [10, 34]. The PEIS-Home looks like a typical bachelor apartment of about 25 square meters. It consists of a living room, a bedroom, and a small kitchen — see Fig. 8. The walls have been constructed to be 1.4 m high, so that the observers can get a bird's eye view of the entire apartment. Several devices —called PEIS for "Physically Embedded Intelligent Systems"— live in the PEIS-Home, and can be connected in different configurations. Devices include mobile robots, a smart fridge with a simple internal manipulator, a visual

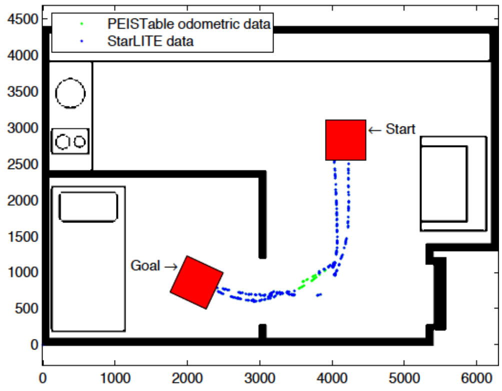

Fig. 9. Odometry and localization data of the PEIS*-Table during navigation in the* PEIS*-home.*

tracking system using ceiling cameras, and many more. Communication and cooperation among PEIS is realized through the PEIS-Middleware, implemented in a run-time portable library which provides a distributed tuple-space over an ad-hoc P2P network.

The PEIS-Table has been incorporated as one of the several PEIS present in the PEIS-Home environment. As such, the PEIS-Table interfaces to the other PEIS by exposing the *functionalities* which it is capable of providing in the home, as well as all information regarding its current state. Such functionalities include navigation, docking to specified points of interest, undocking, the use of its lights, and so on. In addition to providing these functionalities, the PEIS-Table can form ad-hoc coalitions with the other PEIS in the PEIS-Home through the PEIS-Middleware, in order to realize more complex services.

Figure 9 shows how localization of the PEIS-Table is achieved trough collaboration between two members of the PEIS-Ecology during a sample navigation from Start to Goal and back. When moving inside coverage area of the external localization system, the PEIS-Table position is computed by the StarLITE (blue trajectory) and it is fed to the PEIS-Table. When the PEIS-Table loses sight of the tags, the table maintains a dead-reckoning position estimate using its own odometry (green trajectory), and it feeds it to the StarLITE to allow it to more easily recover localization when tags become visible again.

5.3 The PEIS-Table in a Cooperative Task scenario

We now show an experiment in which the PEIS-Table is used as one component in a complex scenario that has been run in the PEIS-Home. This scenario was developed to illustrate how multiple robots and intelligent sensors can be coordinated to obtain a "robotic butler". The scenario develops as follows.

A person enters the PEIS*-Home and sits on the sofa. A stereo camera mounted on the ceiling recognizes the presence of the person and supplies a coarse estimate of his position. A mobile robot equipped with a pan-tilt camera is then dispatched to the person's approximate position to identify the guest, using a face recognition algorithm. In this scenario it is assumed that the system knows the favorite drink of a set of frequent guests. The* PEIS*-Table is sent towards the fridge to fetch the guest's favorite drink. The fridge, equipped with an internal gripper, an internal camera and an actuated door places the drink on the* PEIS*-Table, which has in the meanwhile docked the open fridge. The* PEIS*-Table then navigates towards the person's current position to deliver the drink.*

The scenario is loosely inspired by a test of the RoboCup@Home league of the RoboCup [35] competition. However, whereas a typical approach to this task in RoboCup tends to concentrate functionality on a single robotic platform, the PEIS-Ecology approach leverages the coordination of functionalities provided by multiple PEIS.

Key moments of an example run[2] are shown in figure 10. The run involves several PEIS: the PEIS-Table, an ActivMedia Peoplebot mobile robot equipped with a face tracking and recognition algorithm based on OpenCV [36], a PEIS dedicated to localizing the person through the images fed by the ceiling stereo camera, the fridge with its drink-localization and manipulation capabilities, and an overall controller in charge of coordinating the services of all the other PEIS.

The controller is a script-like program developed specifically for this demo. Among other things, it sequences the tasks of the PEIS-Table and the autonomous fridge, allowing the latter to dock the fridge only once its door has been opened, and allowing the fridge to close the door only once the robot is undocked. The controller also engages the PEIS-Table in a person-following operation if the person moves, by connecting the output of the person tracker to the input of the path planner.

6 CONCLUSIONS

Many observers claim that the inclusion of robotic technologies in everyday environments will be the next inevitable step in the evolution of our homes. If this is the case, we maintain that a substantial role in this development will be played by the design and deployment of robotic furniture, that is, ordinary furniture augmented with robotic technologies. The intention of this paper was to show a concrete instance of this process.

In doing so, we have made three technical contributions. First, we have described the design of a specific piece of

[2] A video is available at http://aass.oru.se/~peis/demonstrator.html #scenario8

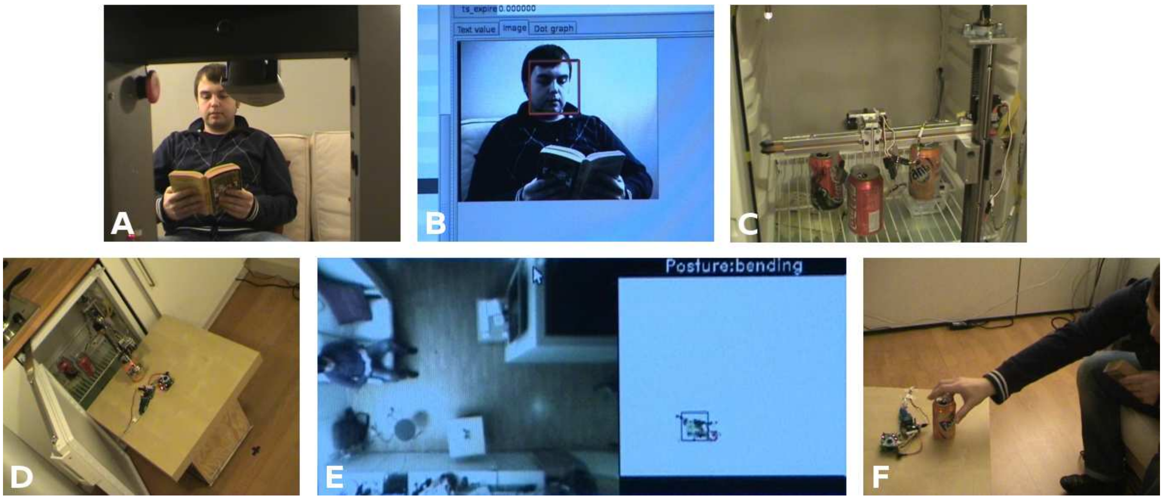

Fig. 10. Snapshots from the execution of the example run: (A, B) the PeopleBot and its on-board pan-tilt camera recognizing the person; (C) the fridge grasping the drink and (D) placing it on the docked PEIS*-Table; (E) the stereo-camera for locating the position of the person in the home; and (F) the user receiving his drink.*

robotic furniture, the PEIS-Table, and have shown examples of its inclusion and use in a smart home. Second, we have discussed the requirements for robotic furniture, and have shown that these may be different from the usual requirements for mobile robots. Third, we have described a set of navigation techniques built from these requirements, and shown how they have been implemented in the PEIS-Table in enough detail to allow reproduction by others.[3] It should be noted that the safe navigation planning is a novel contribution specifically designed for mobile robotic furniture.

While the technical development presented in this paper is specific to the PEIS-Table, we believe that the overall methodology can be applied to most types of robotic furniture, and we hope that it will be of inspiration to other people. Now that the feasibility and added value of robotic furniture has been show as a proof-of-concept, the next important step will be to engage in systematic and long-term user studies. Important questions to be addressed in these studies include whether robotic furniture is really more acceptable than standard robots, and a deeper investigation on the hardware and software requirements for robotic furniture.

ACKNOWLEDGMENTS

The idea of a moving robotic table was initially suggested by Donatella Guarino. Mathias Broxvall, Amy Loutfi, Federico Pecora, Bo-Lennart Silverdhal, and Per Sporrong provided substantial help in realizing this idea. Many thanks to students of the International Master's in Robotics and Intelligent Systems at Örebro University who developed the drink scenario, namely Li Feng, Nikolay Kramarev, Wenjie Lin, Wang Min, Andreas Persson, Tom Ranskär, and Jonas Ullberg.

[3] The full software of the PEIS-Table is available on request.

REFERENCES

[1] H. Graf and R. Schraft, "Robotics home assistant Care-O-Bot: Past - present - future," in *Proc of the Int ROMAN Symp*, pp. 380–385, 2002.

[2] B. Graf, M. Hans, and R. Schraft, "Mobile robot assistants: issues for dependable operation in direct cooperation with humans," *IEEE Robotics and Automation Mag*, vol. 11, no. 2, pp. 67–77, 2004.

[3] E. A. Sisbot, A. Clodic, L. Marin, M. Fontmarty, L. Brèthes, and R. Alami, "Implementing a human-aware robot system," in *Proc of the Int ROMAN Symposium*, pp. 727–732, 2006.

[4] S. Coradeschi and A. Saffiotti, "Symbiotic robotic systems: Humans, robots, and smart environments," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 82–84, 2006.

[5] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Special issue on network robot systems," *Robotics and Autonomous Systems*, vol. 56, no. 10, 2008.

[6] A. Sgorbissa and R. Zaccaria, "The artificial ecosystem: a distributed approach to service robotics," in *Proc of the IEEE Int Conf on Robotics and Automation (ICRA)*, 2004.

[7] T. Heinroth, A. Kameas, G. Pruvost, L. Seremeti, Y. Bellik, and W. Minker, "Human-computer interaction in next generation ambient intelligent environments," *Intelligent Decision Technologies*, 2010.

[8] W. Yu, J.-Y. Lee, H. Ahn, M. Jang, and Y. Kwon, “Design and implementation of a ubiquitous robotic space,” *IEEE Tran on Automation Science and Engineering*, 2009. To appear.

[9] O. Lemaire, K. Ohba, and S. Hirai, “Dynamic integration of ubiquitous robotic systems using ontologies and the RT middleware,” in *Proc of the Int Conf on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2006.

[10] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B. Seo, and Y. Cho, “The PEIS-ecology project: vision and results,” in *Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, pp. 2329–2335, 2008.

[11] N. Tomokuni, B. Kim, T. Tanikawa, K. Ohba, and S. Hirai, “Design of the active caster for the actuation devices of ubiquitous robots,” in *Prof of the Int Symp on Flexible Automation*, 2006.

[12] A. Saffiotti and D. Guarino, “Interactive furniture in a robot ecology,” in *Workshop on Imagining Domestic Interiors*, (Århus, Denmark), 2008.

[13] I. Siio, J. Rowan, N. Mima, and E. Mynatt, “Digital decor: Augmented everyday things,” in *Graphics Interface*, pp. 155–166, 2003.

[14] F. Kawsar, K. Fujinami, and T. Nakajima, “Augmenting everyday life with sentient artefacts,” in *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pp. 141–146, 2005.

[15] A. Sproewitz, M. Asadpour, A. Billard, P. Dillenbourg, and A. Ijspeert, “Roombots–modular robots for adaptive furniture,” in *IROS Workshop on Self-Reconfigurable Robots, Systems and Applications*, 2008.

[16] C. Yu, F. Willems, K. Haller, R. Nagpal, and D. Ingber, “Self-adaptive furniture with a modular robot,” in *Workshop on Imagining Domestic Interiors*, (Årus, Denmark), 2008.

[17] L. Baillie, D. Benyon, S. Bødker, and C. Macaulay, “Special issue on interacting with technologies in household environments,” *Cognition, Technology & Work*, vol. 5, no. 1, 2003.

[18] J. Forlizzi and C. DiSalvo, “Service robots in the domestic environment: a study of the roomba vacuum in the home,” in *Proc of the ACM Conf on Human-Robot Interaction*, pp. 258–265, 2006.

[19] D. Norman, *The Design of Everyday Things*. Basic Books, 1988.

[20] E. Agree and V. Freedman, “Incorporating assistive devices into community-based long-term care,” *Journal of Aging and Health*, vol. 12, no. 3, p. 426, 2000.

[21] A. Clodic, R. Alami, V. Montreuil, S. Li, B. Wrede, and A. Swadzba, “A study of interaction between dialog and decision for human-robot collaborative task achievement,” in *Proc of the Int Symp on Robot and Human Interactive Communication*, pp. 913–918, 2007.

[22] G. Cortellessa, G. Koch-Svedberg, A. Loutfi, F. Pecora, M. Scopelliti, and L. Tiberio, “A cross-cultural evaluation of domestic assitive robots,” in *Proc of the AAAI Fall Symposium on AI and Eldercare*, 2008.

[23] M. Dilich, D. Kopernik, and J. Goebelbecker, “Hindsight judgment of driver fault in traffic accident analysis,” *Transportation Research Record*, vol. 1980, pp. 1–7, 2006.

[24] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon, “A human aware mobile robot motion planner,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[25] T. Bailey and H. Durrant-Whyte, “Simultaneous localisation and mapping (SLAM) – Part II: State of the art,” *Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[26] D. Wolf and G. Sukhatme, “Mobile robot simultaneous localization and mapping in dynamic environments,” *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.

[27] H. Chae, W. Yu, J. Lee, and Y. Cho, “Robot localization sensor for development of wireless location sensing network,” in *IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, pp. 37–42, 2006.

[28] “Player/Stage/Gazebo website.” Internet link. [retrieved 2010-02-22] http://playerstage.sourceforge.net.

[29] G. Ulivi, G. Oriolo, and M. Vendittelli, “Real-time map building and navigation for autonomous robots in unknown environments,” *IEEE T. on Systems, Man, and Cybernetics*, vol. 28, no. 3, pp. 316–333, 1998.

[30] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[31] T. Zhang and C. Suen, “A fast parallel algorithm for thinning digital patterns,” *Comm. of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[32] A. Saffiotti, “The uses of fuzzy logic in autonomous robot navigation,” *Soft Computing*, vol. 1, no. 4, pp. 180–197, 1997.

[33] A. Saffiotti, E. Ruspini, and K. Konolige, “Blending reactivity and goal-directedness in a fuzzy controller,” in *Proc of the IEEE Int Conf on Fuzzy Systems*, pp. 134–139, 1993.

[34] A. Saffiotti and M. Broxvall, “PEIS ecologies: Ambient intelligence meets autonomous robotics,” in *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pp. 275–280, 2005.

[35] “RoboCup@Home.” Internet link. [retrieved 2010-02-22] http://www.ai.rug.nl/robocupathome/.

[36] “OpenCV.” Internet link. [retrieved 2010-02-22] http://opencvlibrary.sourceforge.net.

Enrico Di Lello received the B.S. and M.S. dgrees in computer science and automation engineering from Universita’ degli studi Roma Tre, in 2005 and 2008, respectively. He is currently a Ph.D. student at Katholieke Univeristeit Leuven, Belgium. His major research interests include the application of Dynamic Bayesian Networks to human intent estimation, assembly task monitoring and clincal human motion analysis.

Alessandro Saffiotti is full professor of Computer Science at Orebro University, Sweden, where he heads the AASS Mobile Robotics lab since 1998. He received a M.S. in Computer Science from the University of Pisa, Italy, and a Ph.D. in Applied Science from the Universite Libre de Bruxelles, Belgium. His research interests encompass artificial intelligence, autonomous robotics, and soft computing. He has published more than 100 papers in international journals and conferences and co-edited the book "Fuzzy logic techniques for autonomous vehicle navigation" (Springer, 2001). He is a senior member of IEEE.

AUTHORS' ADDRESSES

Enrico Di Lello
Department of Mechanical Engineering,
Katholieke Universiteit Leuven,
Celestijnenlaan 300C 3000 Leuven, Belgium
email: enrico.dilello@mech.kuleuven.be

Prof. Alessandro Saffiotti, Ph.D.
Academy of Science and Technology,
Örebro University,
70182 Örebro, Sweden,
email: alessandro.saffiotti@aass.oru.se

Received: 2010-02-22
Accepted: 2011-02-16