User-adaptive shared control in a mobility assistance robot based on human-centered intention reading and decision making schemes

George Moustris[1] Milad Geravand[2] Costas Tzafestas[1] and Angelika Peer[3]

Abstract— **Mobility assistance robots (MARs) provide support to elderly or patients during walking. The design of a safe and intuitive assistance behavior is one of the major challenges in this context. Here we present work on two modes of physical Human-Robot interaction; one where the human in is direct contact with the MAR, e.g. by holding some handles, and the other where the human releases the handles whilst the MAR has to follow him/her from the front, i.e. contactless.**

For the first mode, we present an integrated approach for the context-specific, on-line adaptation of the assistance level of a rollator-type MAR by gain-scheduling of low-level robot control parameters. A human-inspired decision-making model, the Drift-Diffusion Model, is introduced as the key principle to gain-schedule parameters and with this to adapt the provided robot assistance in order to achieve a human-like assistive behavior. The MAR is designed to provide a) cognitive assistance to help the user follow a desired path as well as b) sensorial assistance to avoid collisions with obstacles while allowing for an intentional approach of them.

For the second mode, an intention-based assistive controller for allowing the robot to follow a human while moving in the front is analysed. This task is particularly challenging in indoor environments, as there are situations that are undecidable, namely in junctions. We describe a novel local kinodynamic planner which concurrently detects discrete routes and continuous motion paths. An intention recognition algorithm is also detailed, along with tests in a T-Junction.

I. INTRODUCTION

A sufficient motor performance that allows performing physical daily activities is a critical requirement for maintaining mobility and vitality, especially for elderly people and patients. Changes due to aging or disease may result in the limitation of human motor performance, sensing capabilities and cognitive functions, and thus reduce the ability to perform activities of daily living such as walking, transferring or performing personal hygiene. The constantly increasing elderly population, especially in industrialized countries, has led to a strong demand for healthcare specialists and assistive devices. Mobility assistance robots (MARs) can partly cover this demand by providing physical, sensorial, and cognitive assistance.

[1]G.P. Moustris and C.S. Tzafestas are with the School of Electrical & Computer Engineering, National Technical University of Athens, Greece, emails: gmoustri@gmail.ntua.gr, ktzaf@cs.ntua.gr

[2]M. Geravand is with the Chair of Automatic Control Engineering, Technische Universität München, Munich, Germany, email milad.geravand@tum.de

[3]A. Peer is with the Bristol Robotics Laboratory, University of the West of England, Bristol, UK,email: Angelika.Peer@brl.ac.uk

II. CONTEXT-AWARE DECISION MAKING AND SHARED CONTROL

How to adapt the provided assistance depending on the actual context is a major challenge in the controller design of assistive robots. An assistive robot under direct user control can have difficulties guaranteeing acceptable performance and safety due to cognitive, sensorial and physical weaknesses of target users being elderly or disabled people. On the other hand, a fully autonomous system that ignores the user's intention can result in user dissatisfaction and dangerous situations in case of human and robot disagreement. Therefore, a shared control approach allowing human and robot to share the control over resulting actions is typically employed.

In literature most adaptive shared control mechanisms attempt to tune the level of assistance to improve metrics related to the task. Thus, an inherent difficulty lies in deciding on suitable metrics and adaptation strategies such that the overall robot assistance results in a natural behavior to the user. In this context *natural* refers to an intuitive cooperative control scheme that considers human and robot to collaborate as *peers*, meaning that the robot is allowed to make own decisions to online adjust its level of assistance taking current and past information on the user and environment into account. We believe that an intuitive and natural behavior can be achieved if the robot can decide on the provided level of assistance in a similar way to humans. Thus, we formulate the problem of the allocation of control authority as a decision-making problem and employ human-inspired decision-making models. We use the Drift-Diffusion (DD) model, firstly proposed by [1], that describes the decision-making mechanism in humans as a process in which decisions are based on past decisions and the decision criteria are continuously adjusted in order to maximize the reward obtained throughout task execution. Following the principles of the DD model, we propose a mathematical formulation for an integrated control architecture to adapt the parameters of the shared control system of a rollator-type MAR.

The proposed architecture allows to intuitively adapt the short-term a) *cognitive assistance* helping the user to follow a desired path towards a predefined destination, the robot b) *sensorial assistance* to avoid collisions with obstacles and to allow an intentional approach of them, and the more long-term adaptation of the robot c) *physical assistance* based on measured user performance and fatigue.

The adaptive shared control architecture, see Fig. 1 foresees three decision-maker blocks for sensorial, cognitive and physical assistance which are responsible for online adapting the parameters of the admittance controller in order to achieve the desired system behavior. The *Decision on cognitive assistance* block evaluates the planned path towards the goal which is generated by the path planner block, the human navigational intention in form of force and torque applied to the robot handles as well as the actual human performance. The *Decision on sensorial assistance* block uses human input and the information provided by the *Environment state* block, which provides information on the position of obstacles around the robot. Finally, the *Decision on physical assistance* block processes all inputs and adjusts the level of support provided accordingly by manipulating low-level admittance control parameters of the MAR.

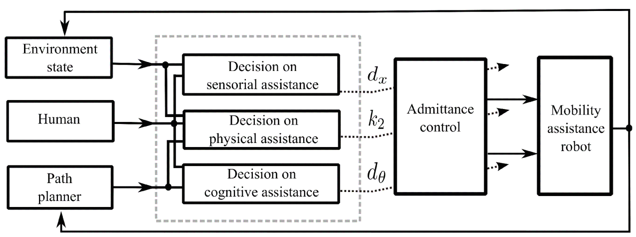

Fig. 1. MAR adaptive shared control architecture.

The effectiveness of the proposed architecture is tested by means of experiments technically validating each of the three aforementioned functionalities of the architecture, see Figure 2 for an example of the autonomous adaptation of the linear damping parameter d_x in case of online adjusting the sensorial assistance as function of the user task performance and human-robot agreement and associated rewards.

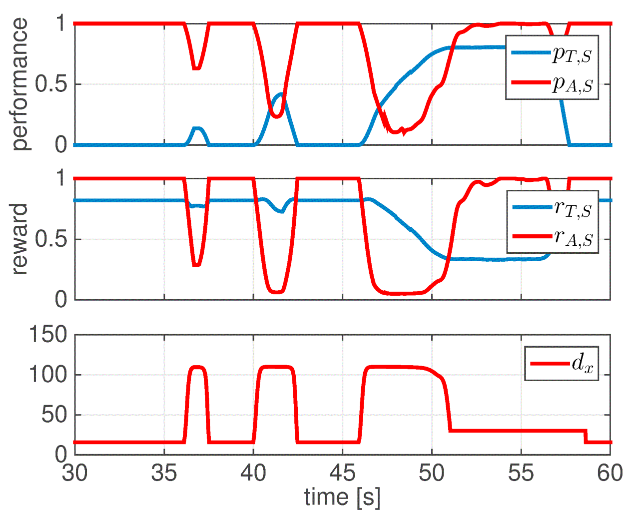

Fig. 2. Results of the sensorial assistance during human-robot cooperation in scenario II.

We further demonstrate the performance of the algorithm with real end-users in a user study with 35 elderly focusing specifically on the sensorial assistance functionality. Obtained results indicate that the required functionalities can be realized with the proposed decision making algorithm showing a general high potential of the proposed adaptive shared control architecture for MARs.

III. Intention based front-following

The main problem with the front-following task, concerns the treatment of eventual *undecidable* situations. This can be exemplified in crossroads (Fig.3 left), where there are distinct routes, completely disjoint from each other. Since the robot resides in front of the user, there might be scenarios where it might be too late for the robot turn correctly, as it may have already moved ”too far” and can't make the turn the user has chosen. Another example is a T-Junction (Fig.3 right). In this case the risk of collision is exacerbated by the fact that, in the time needed to resolve the user intention when the “left” and “right” routes are detected, the robot might be in a “limbo” state, moving further into the junction and either making the routes infeasible or, worse, hitting the wall.

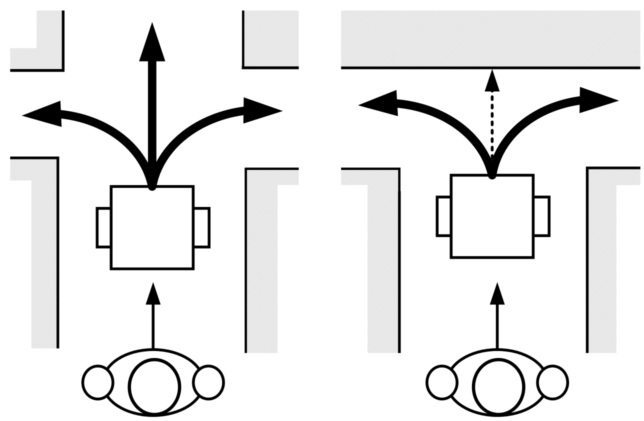

Fig. 3. Example of an undecidable areas; a crossroads (left) and a T-Junction (right).

These two examples reveal a crucial condition; the available routes depend on the environment geometry *as well as* the robot velocity. As such, the planer must take the robot dynamics into account. For example, in the crossroads the robot might be moving *too fast* to make the “left/right” turns, and the only feasible direction would be the “front” route.

In contrast, when moving inside a corridor the available motions form a ”continuum” and the planner must select the best one, according to some scoring function. Identifying undecidable areas is a first key step in the ”front-following” behaviour, with the second one being the *undecidability resolution* through user-intent identification.

An undecidable area can be characterized by the presence of two or more distinct ”routes”. To detect these routes we turn to the notion of *path homototy*. Two paths, with the same starting and ending points, are *path homotopic* if they can be continuously deformed to one another, without colliding with obstacles [2]. The standard definition is too restrictive for *local* path planning because firstly, a goal point is not available and secondly, the path deformation is unconstrained i.e. the resulting paths might not satisfy the differential equations of the robot's motion. By relaxing the condition of the same ending point and imposing constraints on the characteristics of the paths e.g. bounded curvature, we can talk about a more general *path equivalence* [3]. We present a new simple way of producing path equivalence classes in real-time, using a modified dynamic window approach (DWA). Our planner has the advantage of producing geometrically concise classes (called path clusters henceforth) and allowing for the definition of straightforward metrics that convey useful information e.g. cluster span, mean/median path etc.

The DWA [4] is a widely used kinodynamic local planner which searches for collision-free paths in the input space (v, ω). Given a robot velocity tuple (v_R, ω_R), the algorithm samples paths from a window, and simulates them forward in time, selecting an optimal one. The DWA essentially produces *arcs*. For our problem, arcs are a poor candidate since we are not only checking for collision-free paths, but want to calculate "openings" in the surroundings which signify distinct routes of motions. We thus propose to test, instead of arc paths, *arc-line* paths i.e. arcs that are followed by a straight line. These seem more natural a solution for finding possible routes in space, as they can handle turns in a more intuitive way than arcs.

To generate the path, the planner samples an occupancy grid with a fixed spatial resolution from the robot position up to a circle of radius R, centred at the robot, checking for obstacle collision along the way. If the path collides, it is cut off to the collision point (Fig.4). The path is assigned the *minimum cell cost* of the cells it traverses, unless it collides, in which case it has a predefined maximum cost.

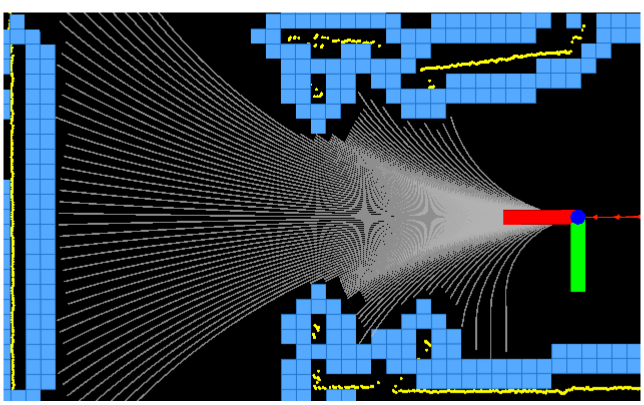

Fig. 4. Path bundle for a T-Junction. Costmap obstacles are cyan, laser points are yellow and white are the AL paths. Robot frame is with thick red-green lines (*x*-*y* resp.).

As we can see from Fig.5, the planner produces two cluster as it approaches the T-Junction. In that moment, the robot has to "signal" the user that it has detected an undecidable area and is reading his/her intention. However, the question arises as to what to do until the intention is resolved. It is evident that the robot must, in parallel, scan its immediate area ahead and create *feasible* motion clusters for that short period of time. To accompany this, we define two scanning circles, called *levels*, with different radii R_{far} and R_{near} (set to $4m$ and $2m$ resp. in our study). The *near level* is seen in Fig.5, comprising a single cluster (in red). Until the robot resolves the user intention, it switches from the *far level* to the *near level* for motion commands. Simultaneously, it enters into the "intention estimation" mode and tries to discern in which *far cluster* the user is heading to. Upon resolution, it discards the *near level* and uses again the *far level* for motion.

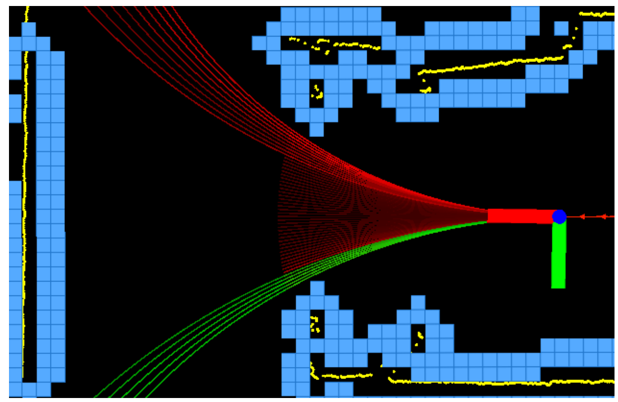

Fig. 5. Far and Near clusters for a T-Junction.

A. Intention Estimation

In the context of the front-following task, intention estimation is referring to the selection of the user-intended *far* cluster, in the presence of undecidability. By observing the user position w.r.t. the robot, we select an angle ϕ_H on the circle of the *far level*. Now consider that there are N *far clusters*. During the intention estimation, the robot assigns scores S_i to the clusters, and increases them by picking the one closest to ϕ_H based on their angle. The selected cluster has its score incremented, by adding a vote. The voting mechanism assigns a lower vote to clusters that are "ahead" than clusters that are "on the side". This has been selected because, if the user wants to promote side clusters, he/she will have to step *away* from his/her normal walking direction, in order to increase the offset. Such a motion is improbable to have been performed by chance and it is most likely a deliberate user action. Hence, when the robot detects it, the "confidence" is high and quickly promotes the selected cluster.

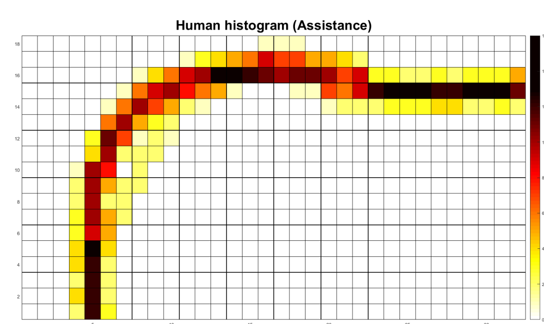

Fig. 6. Histogram of 20 human traces on a left turn in a T-Junction.

After each iteration, the robot selects the top two scoring clusters and compares their scores. If the top score is 50% bigger than the second one, the algorithm terminates and outputs the top cluster. A second condition is that the clusters must have *at least* a predefined number of votes (currently 10). If these conditions aren't satisfied after a *timeout* e.g. $3sec$, it picks the top cluster and exits.

User tests on a T-Junction are seen in 6. Twenty users where asked to perform a left turn on a T-Junction, having the robot following them from the front. The statistical analysis shows the benefit of the assistive mode

ACKNOWLEDGMENT

This work is supported in part by the MOBOT project within the 7th Framework Programme of the European Union, under the grand agreement n. 600796.

REFERENCES

[1] D. M. Egelman, C. Person, and P. R. Montague, "A computational role for dopamine delivery in human decision-making," *J. of Cognitive Neuroscience*, vol. 10, no. 5, pp. 623–630, 1998.

[2] J. Munkres, *Topology*, 2nd ed. Upper Saddle River, NJ: Pearson, Dec. 1999.

[3] R. A. Knepper, S. S. Srinivasa, and M. T. Mason, "An Equivalence Relation for Local Path Sets," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. C. Lin, Eds. Springer Berlin Heidelberg, 2010, no. 68, pp. 19–35.

[4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.