



A Shared-Control Framework for A Human-Robot Front-Following Behaviour in Unknown Dynamic Environments

George Moustris² · Costas Tzafestas^{1,2}

Received: 18 November 2024 / Revised: 10 January 2026 / Accepted: 30 January 2026
© The Author(s) 2026

Abstract

We present a modular framework for enabling a mobile robot to follow a user from the front inside unknown, dynamic and relatively obstacle-laden environments. This behaviour is developed in the context of robotic assistance and rehabilitation for people with mobility impairments. The system incorporates shared-control coupled to an efficient local planner with an intention-reading algorithm recognizing the user intent in undecidable path branches. The components and interconnections are discussed in detail. Field trials with real users are presented, showing that the assistive functionality of the framework encourages users to walk closer to their natural pattern, as compared to an approach with no intelligent features. Preliminary observations suggest that the assistive mode may reduce the cognitive effort required from users. Furthermore, the success rate of the system in identifying user intent is demonstrated to be very high. It is also shown that in certain situations the user-robot pair exhibits complex interaction dynamics, in which their roles are fluid and interchangeable.

Keywords Human-robot interaction · Human following · Shared control · Kinodynamic planning · Intention reading · Local planning

1 Introduction

This work is concerned with the challenges involved in developing a ubiquitous form of Human-Robot interaction; moving in unison i.e. having the robot and the human traverse the environment while being aware of and in close proximity to each other. In such a context, either the robot can follow the user, the user can follow the robot, or have an interchange of roles between the two. Although this behaviour has been vividly investigated by the robotics community Kruse et al. [1], Honig et al. [2], Islam et al. [3] for up-to-date surveys), the vast majority of the literature considers the robot following the user from behind. However, we can

generally distinguish three categories for human-following [4] *behind the user*, *side-by-side* and, *in front of the user*.

Following the user from behind eases the control problem since the user intention can be readily discerned from their trajectory. The most common strategy is *direction following* [2], in which the robot continuously moves to the direction of the user. A second approach involves *following the path* of the user [5], while a third tries to predict the human position, using it as a goal for the robot. Back-following, however, presents some drawbacks. [6] have noticed that when the robot follows the human from behind, the human looks back to see where the robot is. This imposes a cognitive load and causes the user to pay attention to the robot either from curiosity, e.g. see where the robot has gone, or even from fear of the robot bumping onto the user. Further studies have shown [7] that users feel least comfortable when the robot follows them from lateral positions to directly behind.

Recently, the “side-by-side” and “following from the front” problems have been addressed by some researchers. In contrast to “back-following”, the “side-by-side” and “following from the front” tasks face significantly increased difficulty. Specifically, in the former, the user orientation must also be taken into account in order for the robot to stay by the side of the human, while in the latter, prediction of the user’s

✉ George Moustris
gmoustri@mail.ntua.gr

Costas Tzafestas
ktzaf@cs.ntua.gr

¹ School of Electrical & Computer Engineering, National Technical University of Athens, Zographou Campus, Athens 15773, Greece

² Institute of Robotics, Athena Research Center, Artemido 6 & Epidavrou, Maroussi, Athens 15125, Greece

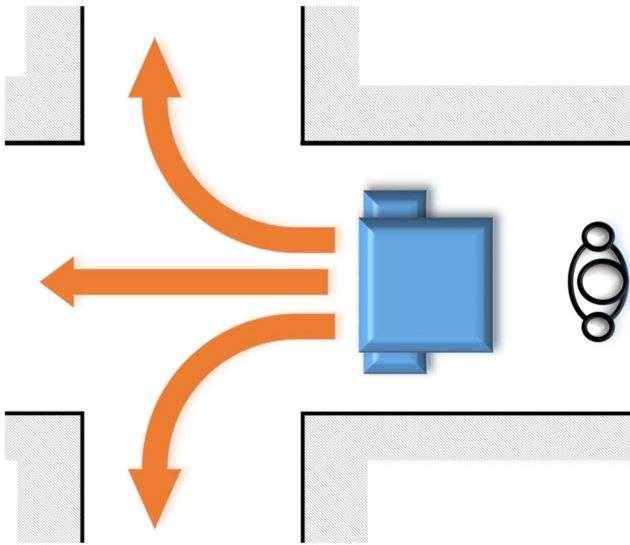


Fig. 1 The fundamental problem of front-following regarding undecidability and user intention: how can the robot discern which turn the user will take, when it hasn't been reached yet?

intention must also be incorporated. To make things worse, in some cases the robot might face *undecidable* situations, requiring user feedback to escape deadlocks. These can be exemplified by crossroads where there are distinct routes, completely disjoint from each other. Since the robot lies in front of the user, there may be scenarios in which it may be too late for the robot to turn correctly, as it may have already moved “too far” into the junction (Fig. 1). In contrast, when moving inside a corridor the available motions form a “continuum” and one has but to select an optimal one, according to some objective function. Identifying undecidable areas is the first key step in this behaviour, with the second one being the *resolution* of such undecidability.

In this work, we focus on having the robot follow the user *from the front*, a behaviour called “*front-following*”. We propose a framework which employs real-time local kinodynamic planning in dynamic environments, detecting discrete, continuous, and undecidable routes. Undecidability is resolved using a user *intent identification* scheme, based

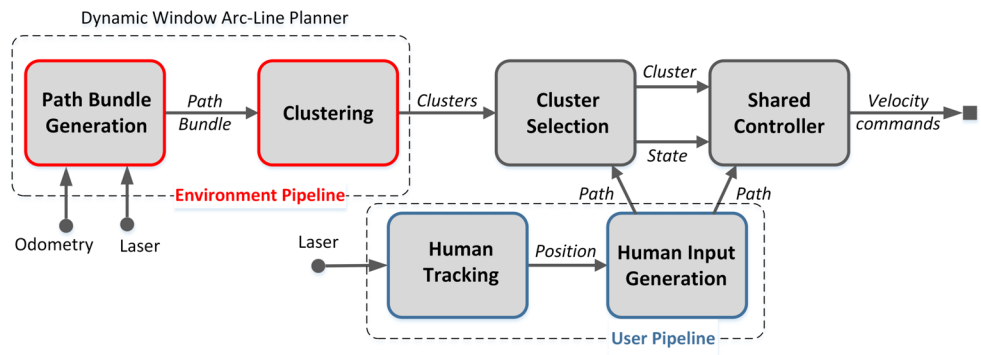
on the user-robot spatio-temporal relation. During normal navigation, the user’s and the robot’s preferred motions are fused based on the surrounding environment’s “narrowness”, using *shared control*. All these tasks are arbitrated by a finite state machine, which transitions between them under the appropriate conditions.

This work stems from a series of research projects, the latest of which is the i-Walk project (<http://www.i-walk.gr>) regarding the development of a modular intelligent robotic rollator for the elderly and people facing motor and/or cognitive impairments. Front-following adheres to the concept of *cognitive Human-Robot Interaction* [8], i.e. non-physical interaction where the robot follows the user from the front in order to oversee them, analyse their gait and provide postural assistance either autonomously or on demand. It can also find applications in rehabilitation robots, providing a similar service. Since the most common form of patient-robot interaction in this context relies on physical manipulation of the device through the handlebars, e.g. [8–11], this excludes patient groups who present with significant upper limb weakness or coordination issues. e.g. major stroke, cerebral palsy, cerebellar syndromes etc.

Other exemplified applications are robots that carry loads e.g. robotic shopping carts, pickers, forklifts, suitcases or even UAVs which always face the user [12].

The general architecture of our system considers both the environment and the user, arbitrating them to produce a safe motion for the robot (Fig. 2). It consists of the *Environment Pipeline*, using our kinodynamic planner to produce available motion clusters, and the *User Pipeline* which processes the human behaviour to infer intent. The two run in parallel and are fused into the *Cluster Selection* module which implements the Finite State Machine, monitoring the system’s state, detecting undecidable situations and resolving them using intent recognition. It produces the desired cluster in which the robot has to move. This is picked up by the *Shared Controller*, which then fuses the user’s and the robot’s desired motion, outputting velocity commands to steer the robot.

Fig. 2 General architecture of the shared-control framework



1.1 Contributions

Preliminary results of our work have been incrementally presented in a series of conference papers [13, 14]. This paper presents a number of significant contributions, namely:

- (a) A path planning and clustering process which produces permissible routes safely guiding the user through narrow, cluttered or high-undecidability areas;
- (b) A user intention recognition algorithm which robustly resolves undecidable situations to smoothly follow user intent;
- (c) Elaboration of a finite-state machine framework which orchestrates the user-robot interaction to enable an autonomous, context-adaptive, shared-control mode of operation;
- (d) Extensive experimental evaluation of the whole system, both under controlled, as well as realistic settings.

In detail, the following significant extensions with respect to our earlier work are presented:

- The clustering process has been extended with new mechanisms to improve performance in cluttered and dynamic environments. In particular, an inter-frame correspondence algorithm is introduced, which matches the derived clusters through space and time. Also, a geometric operation between cluster is described, called “directional slicing”, which removes unwanted paths during turning, leading to smoother motion.
- The intention recognition algorithm has been enhanced by introducing *persistence*; a mechanism to keep the user-selected cluster “selected” in subsequent frames. Furthermore, the scoring mechanism now has a temporal dimension allowing score “decay” through time. This way, the contribution of false positives to the final selection is attenuated, leading to more robust and quick selection process.
- The shared-control framework proposed is formally described by means of a Finite State Machine, which governs the overall operation of the system, encoding all appropriate critical states and events.
- A full series of systematic and extensive experiments has been performed, followed by a statistical analysis of the results which quantitatively validates the performance of our proposed framework enabling more in-depth conclusions.
- A real-life experiment in an uncontrolled, unknown and dynamic environment has also been conducted and is presented in this paper, allowing the observation of the behaviour of the system in real situations, with humans

randomly walking and interacting with the system as it operates in the premises.

The rest of the paper is organized as follows; Sect. 2 surveys work related to the front-following task. Section 3 presents the problem definition and the kinodynamic planner. The human tracking technique and the process of computing the human-selected path, are analysed in Sect. 4. Undecidability detection and resolution is presented in Sect. 5. The shared control algorithm is detailed in Sect. 6 while the experiments, their analysis and a discussion about the results, are given in Sects. 8 and 9.

2 Related Work

The front-following problem has primarily been addressed through anticipatory control strategies, in which the robot infers the user’s motion intent and tracks a virtual target projected in front of them. For instance, Jung et al. [6] use a particle filter over LRF data to estimate the user’s pose and track a forward-projected goal. Similarly, Mi et al. [15] estimate the bisector of the user’s shoulders using a Kinect and apply a Kalman filter for goal prediction, though their system is sensitive to gait variability. This issue is addressed in [16], where multiple LRFs and a nonlinear model predictive controller yield more stable and human-friendly tracking while keeping the robot within a constrained comfort zone.

Pose estimation and filtering are also used in systems such as [4, 17], which apply the nonholonomic human model [18] and unscented Kalman filters for smoother tracking. Shen et al. [17] introduce a sway suppression algorithm to stabilize heading estimation from a 3D torso scan. In contrast, Gai et al. [19] implement a gesture-controlled cart using Kinect but support only straight-line commands. Several other systems leverage deep learning to anticipate user motion. Zhao et al. [20] use LRF and thermal sensors to capture gait and apply a neural network for intent recognition. This work is extended in [21]. Similarly, Gao et al. [22] use a LRF with a Kalman Filter to predict the user gait and pose. They produce a velocity vector field which guides the robot to a target pose ahead, but do not consider environmental constraints.

More recent work explores sophisticated learning-based pipelines for user motion forecasting. [23] use a deep reinforcement learning approach (D4PG) to generate short-term navigation goals based on user motion, which are then tracked via the Timed Elastic Band (TEB) planner [24]. A similar TEB-based planning architecture is employed in [25], where transformer models predict user skeleton and future trajectory from ZED2 depth images. These forecasts are used to determine a target pose two meters ahead. In

[26], a feedforward neural network predicts user position half a second into the future based on the center of gravity derived from Azure Kinect skeleton data, and a PID controller regulates the robot pose accordingly.

Other anticipatory approaches use multi-sensor setups or wearable devices. Cifuentes et al. [27] combine LRF data with wearable IMU sensors to estimate user kinematics and reduce robot-user pose error through inverse kinematics. However, like most of the above, this system assumes free-space motion. In [28], head orientation is tracked to infer intent and maintain proximity, though the system just stops at obstacles and fails under occlusion when the head is seen from obtuse angles. A small number of systems adopt a passive control scheme where the robot is guided by user interaction. In [29], an AR marker is tracked via a monocular camera to estimate pose, and the user effectively steers the robot by changing lateral offset—akin to pushing a cart. A similar control method is used in [13], where the robot velocity is regulated by distance and heading offset.

All previous systems assume no obstacle presence, whereas only a few approaches incorporate environmental awareness. Hu et al. [30] propose a hybrid method that switches between anticipatory and passive control depending on user distance, blending their velocity outputs with obstacle avoidance commands generated via potential fields. Sensors include an RGB-D camera, a LRF and sonar. While the system performs well in confined environments and junctions, passive steering leads to oscillatory robot motion, and no analysis of how obstacle layout affects user trajectory is provided. Nikdel et al. [31] extend anticipatory tracking with collision-aware goal prediction, projecting the goal ahead of the user and adjusting it via a wall-following-like algorithm if it intersects an obstacle. Although it supports recovery after failed turns, this method requires computationally intensive map processing (e.g., morphological filtering and Hough transforms) and is limited to rectilinear environments. Most recently, [32] combine Monte Carlo Tree Search with deep reinforcement learning to generate local goals that avoid both occlusions and obstacles. These are tracked using TEB, but the method has not been validated at junctions or in cases requiring explicit disambiguation of user intent. Its preference for eliminating occlusion over resolving user intent, may override more contextually appropriate path choices.

In summary, most front-following systems assume free-space operation and rely on predictive models to forecast user motion. Passive systems are simpler but shift control burden to the user. Although several anticipatory systems have been augmented with learning-based user models, only a limited number account for obstacle-rich environments, and even fewer support disambiguation of user decisions at intersections. Where environment-awareness is present,

it is often limited in scope, geometry, or scalability. The approach proposed in this work aims to bridge this gap by integrating general-purpose obstacle reasoning with a lightweight mechanism for detecting and resolving user intent at ambiguous decision points.

3 Dynamic Window Arc-Line Planner

In this section we describe the DWAL planner, motivated by the idea that arc-line paths are better suited for navigation in structured environments than plain arc paths.

3.1 Motivation

An undecidable area is characterized by the presence of two or more distinct “routes”. To detect them, at each point we calculate all the feasible local paths the robot can take, for a specified time interval ahead. Collecting these paths, we see if they can be clustered into *path equivalence* classes [33]. In this work we present a new simple way of producing these classes in real-time, using a modified Dynamic Window Approach (DWA) called the *Dynamic Window Arc-Line Planner* (DWAL). Our planner has the advantage of producing geometrically concise classes (called “path clusters”) and allows for the definition of straightforward metrics that convey useful information e.g. cluster span, mean/median path etc.

The DWA [34] is a widely used kinodynamic local planner which searches for collision-free paths in the input space (v, ω) . It simulates paths forward in time for a predefined period T_s , selecting the optimal one according to a scoring function. The time interval defines a temporal window in the sense that the robot can change its velocity in a bounded way, until the planner reiterates the dynamic window with new values. The DWA essentially produces paths of constant curvature i.e. *arcs*.

For our problem, arcs are a poor candidate since we are not only checking for collision but want to calculate “openings” in the surroundings which signify distinct routes of motions; see, for example, Fig.3A. Arcs, seen dashed on the left, collide with the walls and thus discard valid directions of motion. This collision depends on T_s , which essentially determines the forward simulation time interval of the robot’s motion and thus defines the arc length. If T_s is large, the DWA computes longer paths that reach far into the environment. However, their predictive fidelity is limited since the set of possible paths increases as the simulation looks forward in time. Reversely, a small T_s leads to shorter projected paths, providing a more realistic set of available paths, but its predictive capability is weakened. This trade-off between predictive power and fidelity is inherent in

$$\Delta\Theta = 2 \arcsin(\kappa R/2) \tag{5}$$

and using (4),

$$\kappa = 2 \frac{\sin \phi}{R} \iff \phi = \arcsin(\kappa R/2) \tag{6}$$

For the arc-line paths, at the endpoint of each arc (point C in Fig.3B-Left) we attach a straight line segment \overline{CA} which is *parallel to the y-axis* and tangent to the arc at C. This implies that for the arc-line (AL) paths we have the condition that $\Delta\Theta = \pi/2$. From (4) we see that for these paths $\phi = \pi/4$, hence all arcs of AL paths terminate on a line of 45° starting from O.

For consistency of notation we will define ϕ to be the angle between the x -axis and the segment \overline{OE} , for the AL paths as well. To calculate the relation between κ and ϕ for AL paths we have,

$$\left. \begin{aligned} x = R \cos \phi = |OC| \cos \pi/4 \\ |OC| = 2 \frac{\sin \pi/4}{\kappa} \end{aligned} \right\} \Rightarrow R \cos \phi = \frac{1}{\kappa} \tag{7}$$

and thus,

$$\kappa = \frac{1}{R \cos \phi} \iff \phi = \arccos \frac{1}{\kappa R} \tag{8}$$

To produce the *path bundle*, we sample the circle at fixed angle spacing $\Delta\phi$, such that the arc $R\Delta\phi = 0.1m$. The minimum and maximum ϕ are,

$$\phi_{min(max)} = \arccos \frac{1}{\kappa_{lo(up)} R} \tag{9}$$

For on-line mapping of unknown (and dynamic) frontal obstacles, a rolling costmap centred at the robot is used. Using the frontal LRF, a 2D occupancy grid is produced marking each cell with a cost, characterizing it as either *obstacle*, *free space* or *unknown space*. This was implemented in ROS using the *Costmap_2d* package¹, with a cell edge of 0.1m.

The planner simulates each path, sampling it with a fixed spatial resolution (currently set to 0.1m), and checks for obstacle collision in the costmap along the way. If the path collides, it is cut off at the collision point. The path is given the *maximum cell cost* of the cells it traverses, unless it collides, in which case it is set to a maximum cost. Given that the further a cell is from obstacles, the lower its cost, this ensures a minimum clearance of the path from obstacles.

¹ wiki.ros.org/costmap_2d.

3.3 Clustering

To perform clustering of the path bundle, we firstly discard all colliding paths and define two parameters; the *cluster separation* $C_{sep} \in \mathbb{N}$ and the *minimum cluster span* $W_{span} \in \mathbb{R}$. Two paths $P_i, P_j (i > j)$, belong to the same cluster if $i - j \leq C_{sep}$; thus the cluster separation essentially expresses the minimum number of paths between two clusters. This condition is checked for all the paths and produces clusters $C_{k,l} (l > k)$, with P_k being the front cluster path and P_l the back. Using (8),(6), we can associate to each path its curvature κ and its angle ϕ . The *cluster span* $W_{k,l}$ is the length of the chord defined by the angle $\phi_l - \phi_k \equiv \Delta\phi_{k,l}$, and given by,

$$W_{k,l} = 2R \sin \frac{\Delta\phi_{k,l}}{2} \tag{10}$$

To avoid small clusters which could be the result of noise or could signify very tight spaces which the robot could not pass safely, those with a span $W_{k,l} \leq W_{span}$ (set to 0.2m in our trials), are rejected. For each cluster, we select the “optimal” path P_C , having the lowest cost. This corresponds to an angle ϕ_C and curvature κ_C called the *cluster angle and curvature* respectively.

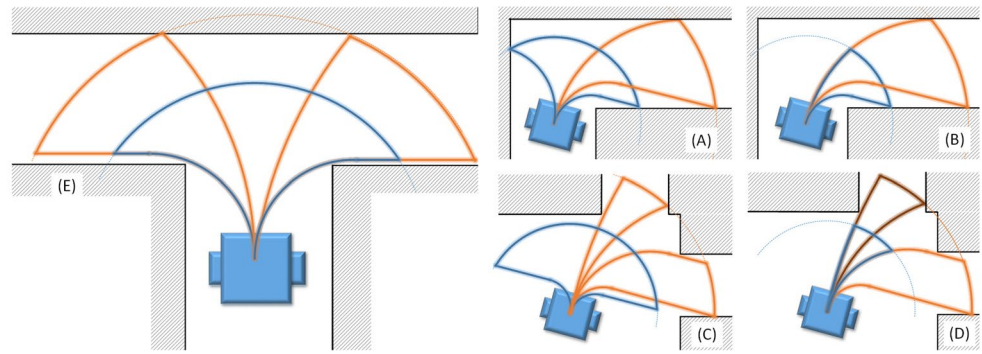
Clustering is also performed using two circle radii (called *levels*). The first is called the *intention level* and sees further into the environment using $R_{far} = 4m$, while the second is the *motion level* with $R_{near} = 2m$ (Fig. 4E). It is crucial to note that these levels serve as abstract categories and may even switch during operation, e.g. the motion level can use R_{far} .

3.4 Directional Slicing

Using multiple levels, the robot has the tendency to make wide turns. This was alleviated by setting the motion cluster to the *near* level using the switching strategy presented in [14]). Further trials revealed that even though the *near* cluster provides a larger span, it also presents directions that are unwanted during turning. For example, in the right turn seen on Fig. 4A, all available directions of the *near* cluster (in blue) pointing *left* towards the outer wall of the turn, are unwanted as they confuse the robot into thinking it can move there. This leads to making *wider* turns and exacerbating the very problem meant to solve. A similar behaviour can be seen in Fig. 4C, where motion directions to the left are unwanted.

To handle this problem, we introduce a cluster operation called “*directional slicing*”. Given a cluster $F_{k,l}$ which the user has selected in the *far* level, and a slicing direction between “*left*” or “*right*”, directional slicing of the cluster

Fig. 4 (LEFT) far (orange) and near (blue) clusters in a T-Junction. (RIGHT) (A+C): using the far cluster (orange) and the near clusters (blue) in turning. (B+D): directional slicing of the clusters



$F_{k,l}$ and the near level R_{near} , produces the path bundle of the said level such that if slicing direction is *right* then its maximum path is P_l and its minimum is either P_n , if another cluster $E_{m,n}$ such that $m < n < k < l$ exists, or P_l if no such cluster exists. The operation of *left* directional slicing is defined in a similar manner.

Examples of “right” directional slicing can be seen in Fig. 4B–4D. In the first, there is only one available cluster in the far level, hence the slicing produces a motion cluster in the near level which spans the maximum path (i.e. the left border) of the far cluster down to the first available non-colliding path in the near level. The resulting “sliced” motion cluster is depicted in blue. In the second case where there are more than one far clusters (Fig. 4D), assuming that the user has selected the left far cluster (in brown color), the slicing produces a near motion cluster which spans from the maximum path of the selected cluster, to the maximum path of the far cluster before it (i.e. on its right). The sliced cluster is again shown in blue. What directional slicing does is remove unwanted directions of motion during turning while providing a wider motion range than the ones offered from the selected far cluster.

3.5 Inter-Frame Correspondence

The clusters are being created in an iterative fashion, matching the update rate of the front LRF. Each cluster has a unique ID number that is propagated over time. As the robot moves around however, the clusters change both in number, size and direction. Complex situations can also emerge, such as the division of a cluster into sub-clusters overlapping with the original one, the elimination of clusters etc. This creates a need for spatio-temporal cluster correspondence between frames, in order to preserve consistency.

The algorithm for this correspondence is based on the minimum distance between the mean cluster angles. Specifically, let at frame T be clusters $C_{k_i,l_i}^T (i = 1...n)$ with mean angles ϕ_i^T , and IDs I_i^T respectively. At frame $T+1$

Algorithm 1 Inter-frame cluster correspondence

```

1: Given:
2: Clusters  $C_{k_i,l_i}^T (i = 1...n)$  at frame  $T$ 
3: Clusters  $C_{k_j,l_j}^{T+1} (j = 1...m)$  at frame  $T+1$ 
4: Distance matrix  $\mathbf{D}$ ,  $D_{i,j} \leftarrow |\phi_i^T - \phi_j^{T+1}|$ 
5: Do:
6: for  $k = 1 : \min(m, n)$  do
7:    $D_{p,r} \leftarrow \min(\mathbf{D})$ 
8:    $I_r^{T+1} \leftarrow I_p^T$ 
9:   discard  $D_{p,r}$ 
10: end for
11: if  $n < m$  then
12:   for  $k = 1 : m$  do
13:     if  $I_k^{T+1}$  has not been assigned then
14:        $I_k^{T+1} \leftarrow k$ 
15:     end if
16:   end for
17: end if

```

there are clusters $C_{k_j,l_j}^{T+1} (j = 1...m)$ with mean angles ϕ_j^{T+1} and IDs I_j^{T+1} . Create the $n \times m$ distance matrix D such that $D_{i,j} = |\phi_i^T - \phi_j^{T+1}|$. Now find the minimum element $D_{p,r}$ in D , corresponding to the clusters C_{k_p,l_p}^T and C_{k_r,l_r}^{T+1} , and assign the cluster ID $I_r^{T+1} = I_p^T$. Following, discard that element and loop over D again, in total $\min(n, m)$ times. If $n \geq m$, that is, there are less or the same number of new clusters at $T+1$, then each has been assigned an appropriate ID. On the other hand, if $n < m$, then there are more new clusters. In this case, start assigning IDs to the new clusters, ranging from *one* up, disregarding the already numbered ones and until all clusters have been numbered. A pseudo-code of the algorithm is given in Algorithm 1.

Since in each iteration the closest pair is discarded, there is no guarantee that each new cluster has the ID of its closest previous one. This holds true only for the first pair.

4 Human Tracking & Input

The robot regulates its behaviour according to the user’s position in order to infer their intention. Consequently, user tracking constitutes a core component of such systems. Many advanced techniques have been proposed in the literature, employing Inertial Measurement Units (IMUs) [27], video cameras [29], and depth sensors such as the Microsoft Kinect [4, 19, 30]. Approaches that infer user intention from torso or head motion, particularly in front-following scenarios, have been investigated by [15] and [28]. Experimental results, however, indicate that such systems are highly sensitive to the walking pattern (in the torso-based case) and to occlusions from oblique viewpoints (in the head-based case). These issues can lead to abrupt or unstable robot motion and, in some cases, collapse the control loop. Moreover, these methods require additional sensing hardware (e.g., depth cameras) and considerable computational resources for skeleton extraction and real-time processing.

Conversely, we adopted a simpler yet robust approach based on a Laser Range Finder (LRF). The LRF is mounted on the rear side of the robot, facing the user, and scans the leg region within a rectangular area termed the Human Interaction Zone (HIZ) [27] (see Fig. 5). This configuration simplifies the control scheme and provides greater robustness to estimation errors compared to approaches that attempt to infer the user’s full-body orientation or velocity. The resulting user position is denoted as (x_H, y_H) . Further implementation details can be found in [39].

The human input module translates the user position into meaningful information regarding the motion of the robot, producing the reference *human velocity* v_H , and the *human angle* ϕ_H . The human velocity is a reference for the *robot velocity*, but is further regulated by the shared controller based on the current system state (see Sect. 6). If (x_H, y_H) is the user position, the human velocity is given by,

$$v_H = \begin{cases} 0 & , \quad x_H > x_0 \\ k_1(x_H - x_0) & , \quad x_2 \leq x_H \leq x_0 \\ v_{walk} & , \quad x_1 \leq x_H \leq x_2 \\ v_R^{max} - k_2x_H & , \quad 0 \leq x_H \leq x_1 \end{cases} \quad (11)$$

where v_{walk} is a predefined walking velocity, $k_1 = v_{walk}/(x_2 - x_0)$ and $k_2 = (v_R^{max} - v_{walk})/x_1$.

According to (11), v_H depends only on the longitudinal distance x_H of the human to the robot. It is defined by three piece-wise linear regions; the *approach region* ($x_2 \leq x_H \leq x_0$), the *walking region* ($x_1 \leq x_H \leq x_2$) and the *collision region* ($0 \leq x_H \leq x_1$). The *approach region* refers to the phase in which the user first engages with the robot, approaching it from a distance. It is the first region where the robot gains velocity to follow the user. When the user is in the walking region, the robot has a constant velocity viz. the *walking velocity* v_{walk} . In this region the robot moves in unison with the user. If the user moves close to the robot, he enters the *collision region* and the robot accelerates up to the maximum velocity v_R^{max} . In our experiments, the following values were used: $v_{walk} = 0.5m/s$, $v_R^{max} = 0.6m/s$, $x_0 = 1.5m$, $x_1 = 0.6m$ and $x_2 = 1.2m$. The selected walking velocity corresponds to the typical gait speed observed in our target population [40].

The *human angle* ϕ_H depends solely on the lateral distance y_H of the user to the robot, viz.

$$\phi_H = \begin{cases} 0 & , \quad |y_H| < \epsilon \\ k_\phi \text{sgn}(y_H)(|y_H| - \epsilon) & , \quad |y_H| \geq \epsilon \end{cases} \quad (12)$$

where ϵ is a deadband and k_ϕ is a constant. The value of ϵ was set to $0.1m$ following tuning sessions with users. Note that the deadband filters the natural walking sway of the user but also changes the sensitivity of the human input system. A narrow deadband might make the system oscillate while a wide one could make it less responsive. The sensitivity of the system can be further regulated by varying k_ϕ .

Using (12), the human selects a moving direction ϕ_H by moving “off-axis” relative to the robot velocity. When the user strafes right (left), a left (right) direction is selected (see the supplementary multimedia file *sv1.mp4*, for a video overview). It also means that in order to turn the robot the user must perform an *unnatural* walking pattern by sidestepping. This magnifies the intentionality of the user as it is unlikely to perform such gait in natural motion. The deadband acts as a noise filter, rejecting normal walking variability.

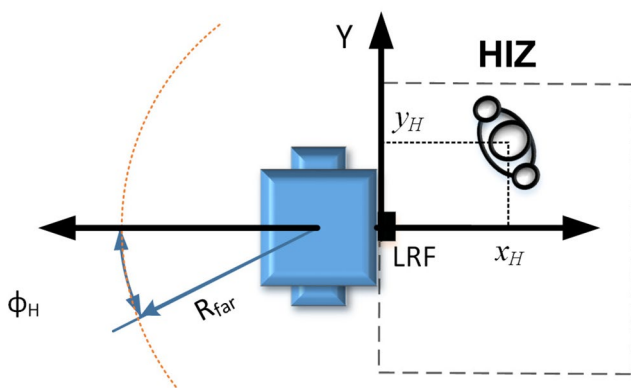


Fig. 5 Illustration of the *human angle* ϕ_H

5 Cluster Selection

The Cluster Selection module is responsible for selecting one of the available clusters for motion. It receives input from the “Environment” and “User” pipelines, fuses them according to a finite state machine, detects undecidable conditions and resolves them through user intent recognition. In general, the following principle is followed:

The user selects the cluster to move in, while the robot selects the path to move on.

Thus, the user makes high-level decisions about the specific route to take when undecidability is detected, and then the robot takes over, calculating continuously the optimal path in the selected cluster. In some situations, robot and user jointly manage the motion path (see Sect. 6).

5.1 Undecidability & Intent

Undecidability emerges when there are distinct motion routes presented to the user/robot pair, and it is impossible for the robot to chose the one the user wants to go, without user feedback. Thus undecidability is a state of the robot. The formalization of undecidability within our framework can be described as follows;

Definition 1 (Undecidability) Let, at frame T , be n clusters available and the user has already selected, and moving in one of them. If at the next frame $T+1$ there are m clusters such that $m > n$, then an undecidable situation has emerged.

When undecidability is detected, the robot enters an *observation* state trying to infer the cluster that the user wants to move in. This *intent recognition* algorithm is based on a cluster scoring mechanism using the human angle ϕ_H , provided by the Human Input module (Eq.(12)). During this state, the clusters start with a zero score. At each time frame, the cluster closest to the human angle is picked up and its score is incremented using a scoring function $S(\phi_H)$, given by,

$$S(\phi_H) = \frac{1}{f} \begin{cases} 1 & , |\phi_H| \leq 0.1\phi_{max} \\ 3 & , |\phi_H| \geq 0.8\phi_{max} \\ 1 + \frac{|\phi_H| - 0.1\phi_{max}}{0.35\phi_{max}} & , otherwise \end{cases} \quad (13)$$

The angle ϕ_{max} is the maximum attainable angle by the robot and f is the update rate of the scoring algorithm. Note that the central clusters are increased by “ $1/f$ ” when selected, while the side clusters increase by “ $3/f$ ”. At each cycle the

top two scores are extracted. When the largest one passes a predefined threshold Θ and it is 50% bigger than the second, then the scoring terminates and the top cluster is selected. If not, scoring continues until a timeout has expired and the top score is chosen. Currently, Θ was set to “3”. This means that border clusters will be selected in *one* second while central ones, in *three*.

Using this scoring function, if the user wants to turn, they perform a strafing motion in order to create an offset from the robot axis and increase the value of y_H . According to Eq.(12), this increases $|\phi_H|$ which ultimately builds up the scores of the side clusters quicker than the central ones. This uneven scoring distribution has been implemented keeping in mind that the strafing motion is an *unnatural* one, and thus if observed, is most likely intentional. Hence, a larger increment expresses higher a confidence on user intentionality.

In the case of straight motion, it is probable that the user fails to realise in time that the robot has encountered undecidability and continues to walk forward, scoring the central cluster. A smaller score increment here expresses the lower confidence on user intentionality.

To make the user perceive the beginning of the “observation” mode more clearly, when undecidability is detected the robot abruptly *halves* its velocity to signify to the user that the state has changed. Hence, the user has but to perceive this sudden drop in the velocity; not its actual value. Importantly, this sudden velocity change is only met during this state transition and cannot be mistaken for any other case.

Furthermore, to make the scoring algorithm more forgiving to false positive scores the scoring algorithm exhibits *decay* with time. Thus, the user-intended cluster must be selected in a persistent manner. The *decay term* is described by,

$$decay = \Theta / fT_D \quad (14)$$

where T_D is the *decay duration*. After T_D secs of decay, the scores have decreased by Θ in total. By tweaking the *decay duration*, one can control the decay velocity.

The update rate f defines how often the scoring mechanism processes the human angle ϕ_H . Since both the score increments (Eq. 13) and the decay term (Eq. 14) scale with $1/f$, the total accumulation and decay over real time are unaffected by f . Thus, the time needed to reach the threshold Θ , and the resulting decision windows, remain unchanged. Higher update frequencies simply provide smoother score evolution without altering the decision timing. In this work we have used $f=10$ Hz to match the Human Input module and ensure responsive behaviour with low computational cost.

5.2 Persistence

Once a cluster is selected, the controller must choose a path within it to follow. However, as clusters evolve during navigation, a mechanism is needed to maintain consistency over time i.e., to preserve the selected cluster across frames. This requires addressing the question: to which cluster at time $T+1$ does the selected cluster at time T correspond?

Building on the principle of minimizing the angular distance (as in the *Inter-frame Correspondence*), we propose two methods to establish this temporal continuity, based on minimizing the distance between the previously followed angle (the “shared angle/path” from Sect. 6) and the current set of clusters. These two methods are termed *soft persistence* and *hard persistence*.

Referencing Fig. 6A-B, consider the shared path at frame T , which by definition belongs to a cluster at that time. At the subsequent frame $T+1$, the system computes the cluster closest to this shared path. In *hard persistence*, the new cluster must explicitly contain the shared path (Fig. 6A). In contrast, *soft persistence* relaxes this condition and simply selects the nearest cluster, even if it does not include the shared path (Fig. 6B). As a result, soft persistence always yields a valid cluster, whereas hard persistence may fail to find one, necessitating fallback behaviour. This makes hard persistence more restrictive and potentially disruptive.

Each strategy serves a different purpose. Soft persistence is used during normal navigation when there is no undecidability, allowing the system to track the user’s motion while aligning with the available motion clusters. Hard persistence is employed during periods of undecidability, where the system actively observes the user to resolve ambiguity. In such cases, since it is unclear which cluster the user intends to follow, the robot attempts to maintain continuity

by persisting with the last selected cluster. If that cluster is no longer valid, it first attempts to switch to the corresponding R_{near} level cluster to increase spatial span. If this also fails, the system forces undecidability resolution by selecting the currently highest-scoring cluster.

6 Shared Controller

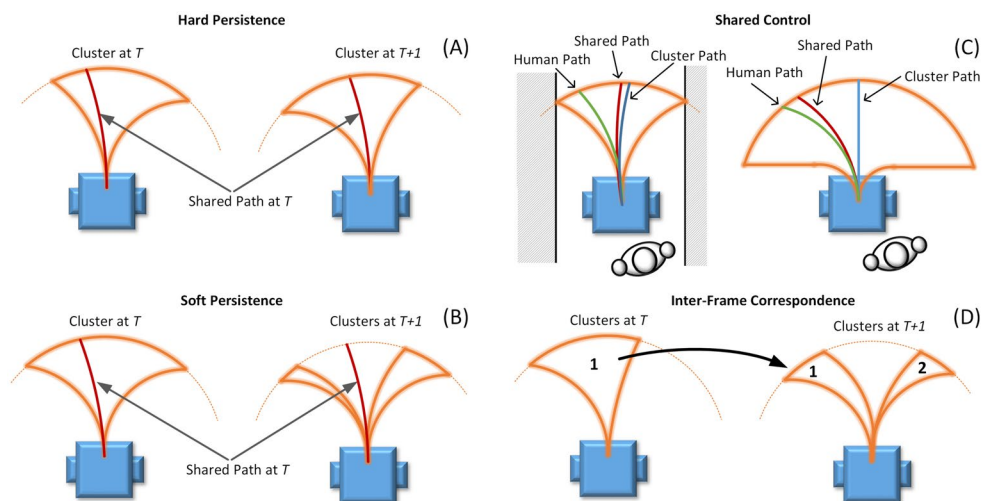
The span of the selected motion cluster expresses the available directions of motion, both for the human and the robot. Thus in open space, the cluster is wide, allowing the user to move the robot freely around. However, in tight spaces motion is restricted, making it more difficult for the human-robot pair to move safely, exacerbating the risk of collision. The smaller the motion cluster is, the less the authority of the human in moving the robot around should be. For example, in very narrow corridors, one can but follow the presented available direction (Fig. 6C). Thus, it would be safer to have the motion planning transferred to the robot. This behaviour can be summarized in the following principle:

In wide spaces motion control is transferred to the human while in narrow spaces motion control is transferred to the robot.

This transfer of authority is implemented by linearly combining the *cluster angle* with the *human angle*, using a parameter “ a ” depending on the cluster width.

Specifically, let ϕ_H , ϕ_R and ϕ_S be the *human angle*, *cluster angle* and the *shared angle* respectively. If the cluster span is W_C , the robot ultimately follows the path corresponding to ϕ_S , given by,

Fig. 6 (A) hard persistence: the selected cluster at $T+1$ includes the shared path. (B) soft persistence: the selected cluster at $T+1$ is the one *1068” ia_version=“0”>closest* to the shared path (in this case, the left one). (C) the shared controller produces a linear combination of the human and cluster paths based on cluster width. In narrow spaces, the *shared path* is closer to the *cluster path*, and the robot has more control. In wider spaces, the *shared path* is closer to the *human path*, allowing the user greater influence. (D) the inter-frame correspondence establishes temporal association between clusters



$$\phi_S = a\phi_H + (1 - a)\phi_R$$

$$a(W_C) = \begin{cases} 1 & , W_C \geq 4 \\ W_C/2 - 1 & , 4 > W_C > 2 \\ 0 & , W_C \leq 2 \end{cases} \quad (15)$$

Using (6), (8), the corresponding path curvature is denoted as κ_S and the commanded robot angular velocity is,

$$\omega_R = \kappa_S v_R \quad (16)$$

The robot velocity v_R is the velocity v_H generated by the human input module, regulated according to the current state. If the system is in an “observing” state i.e. it detects undecidability and starts observing the user in order to resolve it, then the robot velocity is *halved* in order to signal the user the beginning of the observation mode. If the system is in the “idle” state, then the velocity is nulled. In all other states the velocity remains unchanged.

7 Finite State Machine

The Finite-State Machine (FSM) encodes all the probable scenarios which might be encountered during navigation, along with their transitions and events. The entire FSM can be seen in Fig. 7.

It consists of six states and ten boolean events (five events with their complements). The states are composite and comprise two concurrent substates, the *Intention* substate and the *Motion* substate, which reflect the status of the *intention* and *motion* clusters. These substates take their values in {*Normal, Observing, Restricted*} and {*Motion_Far, Motion_Near, Restricted*} respectively. Specifically,

Intention Substate

- Normal:** Intention clusters exist. No undecidability detected
- Observing:** Intention clusters exist. Undecidability detected
- Restricted:** Intention clusters do not exist

Motion Substate

- Motion_Far:** Motion clusters exist and set to the R_{far} level
- Motion_Near:** Motion clusters exist and set to the R_{near} level
- Restricted:** Motion clusters do not exist

Note that the *Intention* clusters are always set to the R_{far} level while the *Motion* clusters switch between R_{far} and R_{near} depending on the current state.

The composite states are of the form “Intention Substate - Motion Substate”. Not all combinations are valid however. The actual states are {*Normal-Motion_Far, Normal-Motion_Near, Observing-Motion_Far, Observing-Motion_Near, Restricted-Motion_Near, Idle*}. The *Idle* state is shorthand for {*Restricted-Restricted*}, when there are no *Intention* and *Motion* clusters available.

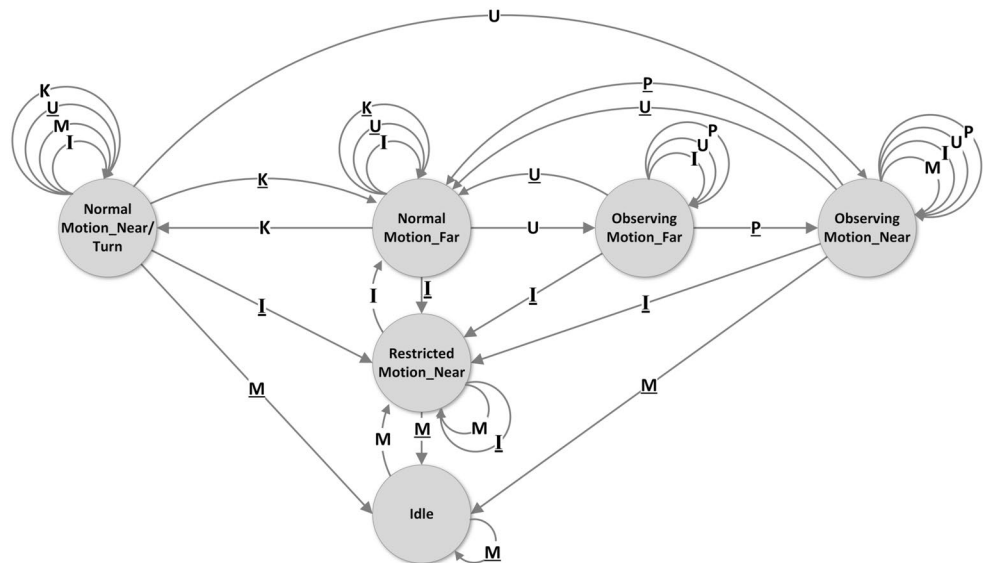
The input to the FSM are ten events which are evaluated inside each composite state. These are actually five true/false events, along with their complements.

FSM Inputs

- U/U:** Undecidability has been detected (true/false)
- K/K:** Robot is turning (true/false)
- P/P:** Hard persistence selects cluster (true/false)
- I/I:** Intention clusters are available (true/false)
- M/M:** Motion clusters are available (true/false)

A *priority* is given to each signal depending on the specific composite state. These can be read from the state transition diagram (Fig. 7) in the way *self-transitions* are encoded. Inputs attached to *inner* self-transitions have a higher priority

Fig. 7 Graph of the Finite state Machine of the system



than *outer* ones and are thus evaluated first. For example, in the $\{Observing-Motion_Near\}$ state, the priorities (from high to low) are $M-I-U-P$. In case one of the inputs leads to another state, evaluation is terminated and the transition is implemented.

8 Experimental Results

Two types of experiments were performed, using a Pioneer P3-DX robot as the experimental platform. The robot was mounted with two Hokuyo UBG-04LX-F01 LRF; one facing forwards, towards the direction of motion, and the second facing backwards, overseeing the user's legs. The first set of experiments was performed in a controlled environment to assess the performance of our framework. Its goal was to observe the gait trajectories of the users with and without the robot and evaluate the behaviour of the user-robot pair; also to infer the impact of the robot on the natural user locomotion pattern. The second type of experiment was performed in a real-life uncontrolled environment, in order to observe the behaviour of the system "in the wild". All algorithms were implemented in the Robot Operating System (ROS Noetic) and ran on a SONY VAIO laptop, placed on the robot.

8.1 Controlled Experiments

Three modes were used in the controlled experiments. The first is the *baseline* mode. Here, the users walked normally

without any robot in front, to establish baseline trajectories which characterize the natural walking pattern, i.e. motion where the user can maintain normal stride, pace, and direction without abrupt deviations imposed by the robot's behavior. The second mode is the *kinematic* mode. This uses a "passive" front-following controller described in [13]. The final mode is the *assistive* mode proposed in this paper.

The experiments consisted of *nine* healthy users moving in a crossroads turning either left, right or just going forward. Thus, each mode consisted of three directions ("left", "right", "straight"). The actual direction was chosen randomly before each trial. For each direction, the user performed *two trials*, summing up to $9 \times 2 = 18$ trials from all users. Consequently each mode had $18 \times 3 = 54$ trials for all directions, and all modes had $54 \times 3 = 162$ trials. Each subject was allowed four test runs before the recordings (two in the kinematic mode and two in the assistive mode), to get acquainted with the system and its functionalities. The experimental area was created inside the 2nd floor of the ECE/NTUA building using cardboard boxes (Fig. 8). The area was pre-mapped in ROS to localize the robot/user trajectories in post-processing. All trials were recorded in ROS and their trajectories calculated using the Adaptive Monte-Carlo Localization algorithm [41].

To compare the trials the experimental field was discretized in a 58×56 square grid, with 0.1m cell edge. For each mode and direction, the histogram was calculated. For example, in the *assistive* mode and *left* direction, if k paths

Fig. 8 UP: human and robot normalized histograms for the *kinematic*, *assistive* and *baseline* modes. Rows represent the *left*, *straight*, *right* directions. All histograms consist of 58×56 square cells, with each cell measuring 0.1 m per side. DOWN: map and photographs of the experimental area

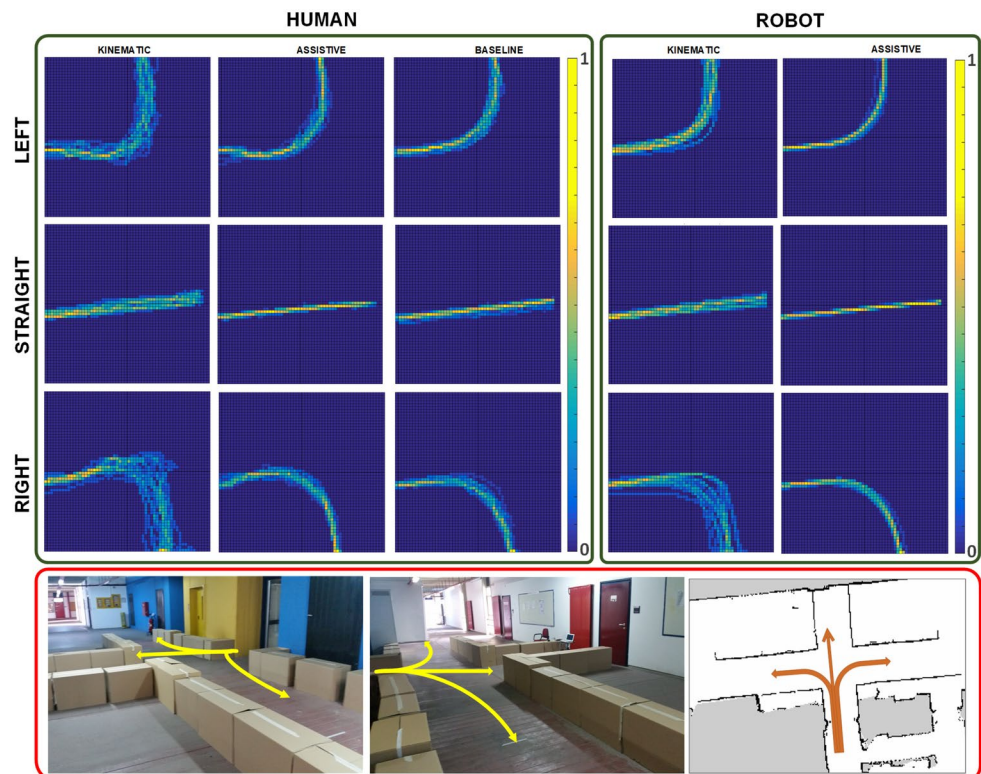


Table 1 Statistical distance of the Human/Robot distributions relative to the Baseline, using the E-statistic

direction	HK-B	HA-B	RK-B	RA-B
<i>left</i>	0.5162	0.4198	0.1110	0.1095
<i>straight</i>	0.1554	0.1070	0.1036	0.0781
<i>right</i>	1.8267	0.5423	0.3255	0.0266

cross the cell $C(i, j)$, then this cell has a value of k . The maximum value a cell can attain is 18. It follows that the histograms express the number of paths traversing each cell. There are in total 15 histograms grouped according to *subject* (Human, Robot), *direction* (Left, Straight, Right) and *mode* (Baseline, Assistive, Kinematic). The histograms can thus be denoted as H_{sm}^d , where “ d ” is the direction, “ s ” the subject and “ m ” the mode. For example H_{HA}^r is the histogram for the *human-assistive-right* trial. The robot and human histograms are presented in Fig.8.

By dividing each cell of a histogram with the sum of all cells, we get a new set of distributions describing the probability of the human/robot being on a specific cell, viz.,

$$P_{sm}^d(i, j) = H_{sm}^d(i, j) / \sum_{i,j} H_{sm}^d(i, j) \tag{17}$$

The distributions $P_{sm}^d(i, j)$ are actual probability density functions, which sum up to one.

To compare the distributions we resort to the *Energy Distance* (ED), which is a measure of statistical distance between two probability distributions [42]. The ED is a relatively recent statistical metric that offers several advantages over more traditional measures. It is defined on arbitrary-dimensional metric spaces, is rotationally invariant and scale equivariant, and, most importantly, captures the geometric relationship between distributions through the

distances in their underlying support space. Unlike divergences such as Kullback-Leibler or Bhattacharyya, which quantify differences based solely on overlapping probability mass, ED also reflects how far apart the distributions lie in space. Thus, two distributions with disjoint support may be considered close or far depending on their spatial proximity, not merely their statistical shape. This property is particularly well-suited for our case, where the distributions represent spatial trajectories; here, closeness must account not only for statistical similarity but also for geometric alignment in the environment.

The ED is zero when the two distributions are identical. To apply it in our results, we consider the relevant concept of *E-statistic* [42], which is applied to statistical samples (see Appendix A for a technical exposition). The *E-statistics* of the Human, Robot distribution relative to the Baseline are presented in Table 1. Essentially this table presents how far or close the Human and Robot paths are to the Baseline.

To test the statistical significance of the findings in Table 1, we calculate the E-Statistic of each trial path relative to the Baseline distribution. We then perform a *t*-test analysis for each mode and each direction separately, as well as cumulatively across all three directions of each mode. The results for each direction per mode and for the aggregated trials (over *ALL* directions) per mode, are presented in Table 2.

The number of cells each histogram covers i.e. the *Cell Count*, can be regarded as a measure of variability and uncertainty in the motion, and consequently of cognitive load when humans are involved. If the histograms are more dispersed, the user covered more space and performed wider manoeuvres to navigate to the goal. The Cell Counts are shown in Table 3 while their relative change with respect to the Baseline histogram count, is presented in Table 4.

Table 2 Two-tailed unpaired t-tests comparing kinematic–Baseline (HK–B, RK–B) and assistive–Baseline (HA–B, RA–B) E-statistics for human (H) and robot (R) path distributions, shown overall and per movement direction

Subject	Mean±SD		n	df	t	p
<i>ALL</i>	•K-B	•A-B				
H	1.09±1.36	0.46±0.10	53	59	3.84	0.0003**
R	0.42±0.21	0.12±0.01	53	55	4.71	0.00002**
<i>LEFT</i>						
H	0.74±0.24	0.54±0.07	18	27	1.52	0.139
R	0.30±0.04	0.18±0.01	18	26	2.46	0.021*
<i>RIGHT</i>						
H	2.24±1.90	0.65±0.09	17	18	4.46	0.0002**
R	0.77±0.41	0.09±0.00	17	16	4.32	0.0005**
<i>STRAIGHT</i>						
H	0.36±0.10	0.19±0.02	18	23	2.10	0.047*
R	0.22±0.04	0.10±0.00	18	18	2.60	0.018*

† The prefix “•” indicates that K-B and A-B refer to the subject in each row (e.g., HK-B/HA-B for H, and RK-B/RA-B for R)

* statistically significant ($p < 0.05$)

** statistically highly significant ($p < 0.001$)

Table 3 Cell counts for human, robot, and Baseline histograms per movement direction (*left, straight, right*), and their relative percentage change between the kinematic and assistive modes

dir.	HK	HA	$\Delta H\%^1$	RK	RA	$\Delta R\%^2$	B
<i>l</i>	476	330	-30.6	327	205	-37.3	280
<i>s</i>	287	136	-52.6	234	119	-49.1	233
<i>r</i>	568	283	-50.1	441	188	-57.3	268

¹ $\Delta H\% = 100(HA - HK)/HK$, ² $\Delta R\% = 100(RA - RK)/RK$

Table 4 Relative change (%) in human and robot histogram cell counts with respect to the Baseline, for the kinematic and assistive modes across movement directions

dir.	$\Delta HKB\%^1$	$\Delta HAB\%^1$	$\Delta RKB\%^1$	$\Delta RAB\%^1$
<i>l</i>	70.0	17.8	16.7	-26.7
<i>s</i>	23.1	-41.6	0.4	-48.9
<i>r</i>	111.9	5.6	64.5	-29.8

8.2 Real-Life Experiment

In this type of experiment, a single user was asked to follow an 8-shaped path, marked on a map of the 2nd floor of the ECE/NTUA building (Fig.9(UP)). See the supplementary multimedia file *sv2.mp4* for the experiment video). The premises consist of two long parallel corridors connected in various points. As the user progresses on the path, he is expected to take the turn A. If the turn is missed, the user continues and makes the turn from the other side. The goal is to have the user move continuously inside the prescribed area. The entire experiment was performed once, unrehearsed and the environment was totally uncontrolled. As a consequence, during the trial random people walked by the robot, doors were opened blocking the way etc. The

trial lasted for *940sec*, and the robot travelled some *425m*. Off-line analysis of the recorded data revealed *49 instances* where the user had to select a cluster of movement (undecidability detection), in order to stay on the path. From those 49, the user *missed 5*. However, from those 5, only 3 are attributed to a failure of the front-following algorithm. In the remaining two, the user chose the wrong route (e.g., Turn B in Fig. 9), but the controller correctly followed the user's actual path. Across all 49 decisions, this yields an intention recognition *success rate of 46/49 = 93.8%*, indicating robust tracking of user intent.

9 Discussion

Analysing the Controlled Experiments, a first conclusion drawn for Table 1 is that the Human Assistive (*HA*) paths are closer to the Baseline (*B*) paths than the Human Kinematic (*HK*) ones. This finding means that the shared-control framework proposed in this paper seems indeed to provide considerable assistance to the user when navigating and moving along with the robot, enabling them to walk more

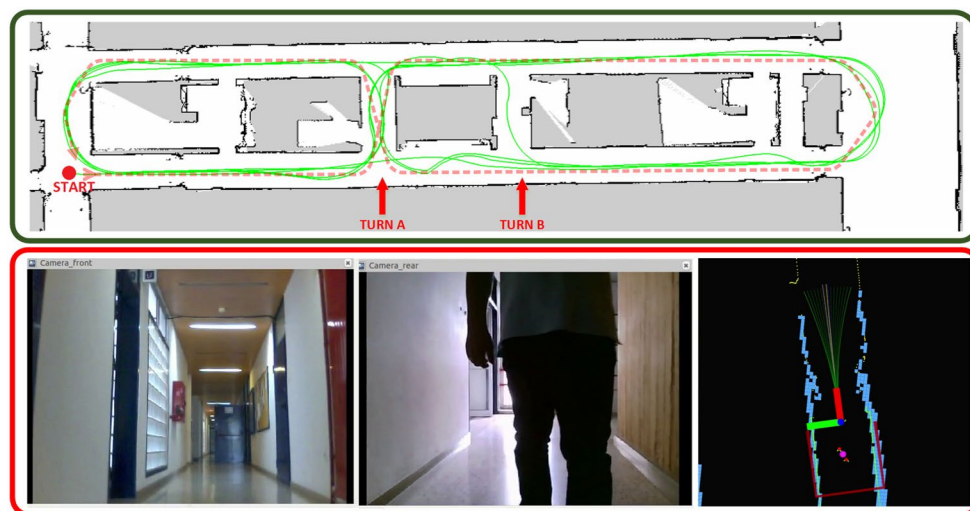


Fig. 9 (UP) map of the real-life experiment. The 8-shaped track is seen in red dashed line. The actual robot path is seen in solid green. (DOWN) screenshot of the experimental video. The views from the two cameras (front-rear) are observed on the left, while on the right the workings of the front-following algorithm are visualized. The robot is depicted as the red/green (*x/y*) frame, with *red* facing forwards.

Obstacles are marked as *cyan* cells. The available cluster of motion produced by the DWAL algorithm is shown as *green* arcs, stemming from the robot origin, towards the *x-axis*. The user is shown as the *pink* dot, with the legs shown as *red* ones. The human interaction Zone is visible as a *dark red* rectangle. The *yellow* dots are the laser scans

Table 5 E-statistics comparing human (HA) to robot (RA) and to Baseline (B) histograms per movement direction (left, straight, right) in the assistive mode. $\Delta\%$ shows the relative difference between HA-RA and HA-B

direction	HA-RA	HA-B	$\Delta\%$
<i>left</i>	0.8547	0.4198	103.60
<i>straight</i>	0.0611	0.1070	-42.90
<i>right</i>	0.4545	0.5423	-16.19

naturally while a robot is front-following. The non-trivial values of the HA-B distances however, reflect the fact that the user must deviate from their normal walking pattern to resolve eventual undecidabilities.

Another interesting point is that the Robot Kinematic paths (RK) are closer to the Baseline than the Human-Kinematic ones (HK). Again, this reconfirms the findings in [13] and [14] in which the users tend to “drive” the robot to the path *they* would have taken.

It is also worth noting that the Robot Assistive paths (RA) are the closest ones to the Baseline. This can be partially attributed to the corridors in the experiments being about 2m wide, making the shared controller pass the control to the robot, per Eq. (15). This in turn, automates the navigation paths of the robot employing the DWAL planner, thus presenting consistency during the experiments. Consequently, it appears that the DWAL planner produces paths close to the ones naturally taken by the users, for the specific experimental settings.

Reading Table 3, we see that the *assistive* controller reduces the amount of traversed cells by the user (HA cell count) by approximately 30% (*Left*) to 50% (*Right*), relative to the *kinematic* case (HK cell count). Moreover, the users cover relatively few more cells in the assistive mode compared to their normal walking baseline paths (Table 4, second column, from 5.6% to 17.8%, in the right/left directions resp). In contrast, the kinematic controller leads to counter-intuitive locomotive patterns, making the users cover significantly more cells in their attempt to “drive” the robot (from 70% to 111%, in the left/right directions resp.).

Regarding the robot paths, Table 3 shows that the proposed *assistive* controller not only produces more compact paths compared to the *kinematic* case (-37% cells *Left*, -57% cells *Right*), but these paths are more consistent than the *baseline* paths, covering noticeably fewer cells (approx. 30%, Table 4, last column).

A last notable comment for the Controlled Experiments, concerns the trials in the *Straight* direction. In the assistive mode, the user walks closer to the baseline than in the kinematic mode (Table 1, HA-B, HK-B distances), whilst also presenting less variability than both (52% less cells than HK -Table 3, $\Delta H\%$ - and 42% less cells than B-Table 4, $\Delta H A B\%$). This is matched by a similar decrease of the robot’s cell count (49% reduction - Table 4, $\Delta R A B\%$). This

pattern however, is not present in the “left/right” trials. To investigate this further, we calculated the distance of the HA distribution to the Robot-Assistive distribution (RA), presented in Table 5 and juxtaposed to the HA-B distances.

We see that in the *Straight* assistive case, the users instead of walking closer to their natural baseline pattern, they moves more similarly to the robot’s paths ($\Delta\%(H A - R A)(H A - B) = 42.9\%$). This, however, is not seen in the *Left* and *Right* cases. A possible explanation for this is that in the *Straight* case the users actually *follow the robot*, instead of the other way around. If so, it is logical to see the human paths more similar to the ones taken by the robot. However, since the robot by design follows the user, this reveals a complex human-robot interaction during front-following in which the roles of the leader/follower might interchange or are ambiguous, depending on the situation. This finding opens up a new interesting direction for further research regarding such a complex behaviour.

Evaluating the Real-Life experiment, the overall assessment is that the proposed framework performs satisfactory even in unknown dynamic environments. Even so, scrutinizing the three failures, we have found that in the *two*, the user tracking module produced spurious estimates of the user position, resulting in the robot taking the wrong turn. The last failure was due to the planner producing an unsafe path, attributed to a human moving rapidly in front of the robot while it was turning, forcing it to collide with the wall.

The limitations of our system are clearly demonstrated by the failure modes discussed earlier, highlighting the necessity for more robust user estimation algorithms and safer planning in fast-paced dynamic environments. Additionally, in scenarios with high obstacle density, such as very crowded places, the clustering algorithm may generate numerous feasible motion or intention clusters. This abundance of clusters can overwhelm the user, making it difficult for them to accurately select the intended cluster based on their position relative to the robot.

Furthermore, in such crowded environments, the complex dynamics of the crowd pose a significant challenge to the stability of the generated clusters. The “openings” within a moving crowd may appear and disappear unpredictably, resulting in chaotic and inconsistent cluster formations. This instability complicates the process of navigating through the crowd and selecting the appropriate cluster, necessitating further advancements in our system to better handle these dynamic conditions.

Finally, it should be noted that several parameters of the proposed framework, including the deadband widths, decay rate, scoring threshold, and motion cluster span, were selected empirically based on pilot experiments and prior experience with front-following navigation. These parameters are not claimed to be theoretically optimal, but were

chosen to ensure stable and predictable behaviour across the evaluated scenarios. As with most shared-control systems, their effective values may depend on user walking style, population characteristics, and environmental structure, and may therefore require adaptation when deployed in different contexts.

10 Conclusion

We have presented a modular framework for enabling a mobile robot to follow a user from the front, navigating inside a dynamic, unknown and obstacle-laden environment. Our approach facilitates the coupling between rapid user-intention reading in undecidable situations, integrated within a shared-control framework based on an efficient, real-time local planner. The proposed system presents the advantage of integrating both human-aware and environment-aware processes under one modularized framework, enabling a more natural, fluent and safe human-robot motion coupling, even in tight spaces.

Field trials with real users showed that the assistive functionality leads the users to walk closer to their natural walking paths, compared to a “passive” front-following controller, with no intelligent features. Further analysis showed that in certain situations the user-robot pair exhibits complex interaction dynamics, in which their roles as leader and follower is fluid and perhaps interchangeable.

Our approach can find numerous applications (rehabilitation, mobility assistance, etc) and serves as a first attempt to formalize this task, containing many degrees of freedom which can be tuned, or modules which can be changed altogether. Hopefully, it will serve as a fertile ground for new and interesting ideas

Appendix A Details on the E-Statistic

Let $X = [x_1, x_2] \in [1, 2, \dots, 58] \times [1, 2, \dots, 56]$, be a 2D random vector. For each histogram H_{sm}^d , we have the respective pool of observations X_1, X_2, \dots, X_n , where each observation contributes *one* account in the histogram. For example, if the cell (i, j) of the histogram has the value k , then there are k observations $X_* = [i, j]$. The total number of observations is $n = \sum_{i,j} H_{sm}^d(i, j)$. If histogram H_1 comprises n observations $\{X_1, X_2, \dots, X_n\}$ of X and histogram H_2 comprises m observations $\{Y_1, Y_2, \dots, Y_m\}$ of Y , then their averages are,

$$\begin{aligned}
 A &= \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \|X_i - Y_j\| \\
 B &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \|X_i - X_j\| \\
 C &= \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m \|Y_i - Y_j\|
 \end{aligned}
 \tag{A1}$$

Their E-statistic is then defined as,

$$E_{n,m}(X, Y) = 2A - B - C
 \tag{A2}$$

Since each observation enters the sum as many times as its respective frequency in the histogram, if X^t, Y^t are the *non-identical* observations of X and Y resp. , i.e. $X^t_i \neq X^t_j, \forall i \neq j$, and n^t, m^t are their respective numbers, then the sums can be transformed to,

$$\begin{aligned}
 A &= \frac{1}{nm} \sum_{i=1}^{n^t} \sum_{j=1}^{m^t} H_1(X^t_i)H_2(Y^t_j) \|X^t_i - Y^t_j\| \\
 B &= \frac{1}{n^2} \sum_{i=1}^{n^t} \sum_{j=1}^{n^t} H_1(X^t_i)H_1(X^t_j) \|X^t_i - X^t_j\| \\
 C &= \frac{1}{m^2} \sum_{i=1}^{m^t} \sum_{j=1}^{m^t} H_2(Y^t_i)H_2(Y^t_j) \|Y^t_i - Y^t_j\|
 \end{aligned}
 \tag{A3}$$

where $H_1(X^t_i) = H_1(x^t_{1i}, x^t_{2i}), X^t_i = [x^t_{1i}, x^t_{2i}]$ (similarly for H_2 and Y^t observations). It is easy to see that the maximum number of unique observations is the number of cells in the histograms. Thus, Eq. (A3) can be written in a way which systematically sums over the histogram cells, viz.

$$\begin{aligned}
 A &= \frac{1}{nm} \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} H_1(i, j)H_2(p, q) \|[i, j] - [p, q]\| \\
 B &= \frac{1}{n^2} \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} H_1(i, j)H_1(p, q) \|[i, j] - [p, q]\| \\
 C &= \frac{1}{m^2} \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} H_2(i, j)H_2(p, q) \|[i, j] - [p, q]\|
 \end{aligned}
 \tag{A4}$$

where in our case $N=58, M=56$ cells. Taking into account Eq. (17) and the fact that $n = \sum_{i,j} H_1(i, j), m = \sum_{i,j} H_2(i, j)$, then it follows that,

$$\begin{aligned}
 A &= \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} P_1(i,j)P_2(p,q) \|[i,j] - [p,q]\| \\
 B &= \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} P_1(i,j)P_1(p,q) \|[i,j] - [p,q]\| \quad (A5) \\
 C &= \sum_{i,j=1}^{N,M} \sum_{p,q=1}^{N,M} P_2(i,j)P_2(p,q) \|[i,j] - [p,q]\|
 \end{aligned}$$

where P_1, P_2 are the probability distributions of the respective histograms.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s12369-026-01376-0>.

Author Contribution GM contributed to the theoretical investigation and implementation of the work. Also prepared the manuscript. CS contributed to the theoretical investigation, oversaw the development and the manuscript preparation.

Funding Open access funding provided by HEAL-Link Greece. This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project i-Walk, code: T1EDK-01248/MIS: 5,030,856).

Data Availability Data sets generated during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics Approval This was an observational study of healthy volunteers who consented to participate. No personal data were collected and no personal identifiable data were produced. The study did not involve an intervention or any kind of physical interaction with the robot. Following initial consultation with the National Technical University's Ethics Committee of Research, our study was exempted from ethics review.

Competing Interests The authors declare no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Kruse T, Pandey AK, Alami R, Kirsch A (2013 December) Human-aware robot navigation: a survey. *Robot Auton Syst* 61(12):1726–1743

2. Honig SS, Oron-Gilad T, Zaichyk H, Sarne-Fleischmann V, Olatunji S, Edan Y (2018) Towards socially aware person-following Robots. *IEEE Trans Cognit Dev Syst* 1–1
3. Islam MJ, Hong J, Sattar J (2019 December) Person-following by autonomous robots: a categorical overview. *The Int J Robot Res* 38(14):1581–1618. (Publisher: SAGE Publications Ltd STM)
4. Ho D, Hu JS, Wang JJ (2012 July) Behavior control of the mobile robot for accompanying in front of a human. In 2012 IEEE/ASME Int. Conf. Advanced Intell. Mechatronics (AIM), pp 377–382
5. Katz D (2016) Development of algorithms for a human-following robot equipped with kinect vision and laser sensors in an unknown indoor environment with obstacles and corners (Unpublished doctoral dissertation). Ben-Gurion University of the Negev, Faculty of Engineering Sciences.
6. Jung EJ, Yi BJ, Yuta S (2012) Control algorithms for a mobile robot tracking a human in front. In 2012 IEEE/RSJ Int. Conf. Intell. Robots and Syst. (IROS), pp 2411–2416
7. Maehara K, Fujinami K (2018 August) Psychological effects on positional relationships between a person and a human following robot. In 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pp 242–243
8. Cifuentes CA, Frizzera A (2016) Human- robot interaction for assisting human locomotion. In: Cifuentes CA, Frizzera A (eds) *Human-robot interaction strategies for walker-assisted locomotion*. Springer International Publishing, Cham, pp 17–31
9. Chuy OY, Hirata Y, Wang Z, Kosuge K (2007 October) A control approach based on passive behavior to enhance user interaction. *IEEE Trans Robot* 23(5):899–908. Conference Name: IEEE Transactions on Robotics
10. Grondin SL, Li Q (2013 June) Intelligent control of a smart walker and its performance evaluation. In 2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR), pp 1–6. (ISSN: 1945–7901) <https://doi.org/10.1109/ICORR.2013.6650346>
11. Jiménez MF, Monllor M, Frizzera A, Bastos T, Roberti F, Carelli R (2019 June) Admittance controller with spatial modulation for assisted locomotion using a smart walker. *J Intell Rob Syst* 94(3):621–637
12. Nagy G, Vaughan R (2017 September) Flying face engagement: aligning a uav to directly face a moving uninstrumented user. In Abstract in Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'17), Vancouver, BC, Canada
13. Moustiris GP, Dometios A, Tzafestas CS (2015) User front-following behaviour for a mobility assistance robot: a kinematic control approach. In 8th international conference on integrated modeling and analysis in applied control and automation, imaaca 2015, pp 142–149
14. Moustiris GP, Tzafestas CS (2016 June) Assistive front-following control of an intelligent robotic rollator based on a modified dynamic window planner. In 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (biorob), pp 588–593
15. Mi W, Wang X, Ren P, Hou C (2016) A system for an anticipative front human following robot. *Proc Int Conf Artif Intel Robot Int Conf Automation, Control Robot Eng* (6):pp. 4:1–4. New York, NY, USA: ACM
16. Sekiguchi S, Yorozu A, Kuno K, Okada M, Watanabe Y, Takahashi M (2020 September) Human-friendly control system design for two-wheeled service robot with optimal control approach. *Robot Auton Syst* 131:103562
17. Shen T, Afsar MR, Zhang H, Ye C, Shen X (2020 December) A 3D computer vision-guided robotic Companion for non-contact human assistance and rehabilitation. *J Intell Robot Syst* 100(3):911–923

18. Arechavaleta G, Laumond J-P, Hicheur H, Berthoz A (2008 August) On the nonholonomic nature of human locomotion. *Auton Robots* 25(1–2):25–35
19. Gai S, Jung EJ, Yi BJ (2013 October) Mobile shopping cart application using kinect. In 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp 289–291
20. Zhao X, Zhu Z, Liu M, Zhao C, Zhao Y, Pan J, Wu C (2020) A smart robotic walker with intelligent close-proximity interaction capabilities for elderly mobility safety. In: *Front. Neurobot.*, vol 14. Frontiers, Publisher
21. Chongyu Z, Wenzhi G, Rongwei W, Wang Z, Wu C (2022 May) Deep learningdriven front-following within close proximity: a hands-free control model on a smart walker. In 2022 International Conference on Robotics and Automation (ICRA), pp 812–818. <https://doi.org/10.1109/ICRA46639.2022.9811910>
22. Gao Z, Wang Z, Saint-Bauzel L, Ben Amar F (2025 February) Large workspace frontal human following for mobile Robots utilizing 2D LiDAR. *J Intell Robot Syst* 111(1):25. doi: <https://doi.org/10.1007/s10846-025-02225-4>. Retrieved 2025-07-4, from
23. Nikdel P, Vaughan R, Chen M (2021 May) LBGP: learning based goal planning for autonomous following in front. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp 3140–3146. (ISSN: 2577-087X)
24. Rösmann C, Feiten W, Wösch T, Hoffmann F, Bertram T (2013) Efficient trajectory optimization using a sparse model. In 2013 european conference on mobile robots, pp 138–143
25. Mahdavian M, Nikdel P, TaherAhmadi M, Chen M (2023 May) STPOTR: simultaneous human trajectory and pose prediction using a non-autoregressive transformer for robot follow-ahead. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pp 9959–9965. Retrieved 2025-06-29, from: <https://doi.org/10.1109/ICRA48891.2023.10160538>
26. Wang A, Makino Y, Shinoda H (2024 Oct) Real-time person-following robot: front-following using human motion prediction. In: *Proceedings of the 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*; 3851–3856. Retrieved 2025-06-29, from: <https://doi.org/10.1109/SMC54092.2024.10831013>
27. Cifuentes CA, Frizzera A, Carelli R, Bastos T (2014 October) Human-robot interaction based on wearable IMU sensor and laser range finder. *Robot Auton Syst* 62(10):1425–1439
28. Scheidegger WM, de Mello RC, Sierra M, Jimenez SD, Múnera MF, Cifuentes MC, Frizzera-Neto CA, A. (2019 June) A novel multimodal cognitive interaction for walker-assisted rehabilitation therapies. In 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR), pp 905–910. (ISSN: 1945-7898)
29. Tominaga J, Kawachi K, Rekimoto J (2014) Around me: a system with an Escort robot providing a sports Player's self-images. *Proc 5th Augment Hum Int Conf* pp 43:1–43:8. New York, NY, USA: ACM
30. Hu JS, Wang JJ, Ho DM (2014 April) Design of sensing system and anticipative behavior for human following of mobile Robots. *IEEE Trans Ind Electron* 61(4):1916–1927
31. Nikdel P, Shrestha R, Vaughan R (2018 May) The hands-free push-cart: autonomous following in front by predicting user trajectory around obstacles. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp 1–7
32. Leisiazar S, Park EJ, Lim A, Chen M (2023 October) An MCTS-DRL based obstacle and occlusion avoidance methodology in robotic follow-ahead applications. In 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 221–228. <https://ieeexplore.ieee.org/document/10342150/>. Retrieved 2025-06-29, from ISSN: 2153-0866
33. Knepper RA, Srinivasa SS, Mason MT (2010) An equivalence relation for local path sets. In: Hsu D, Isler V, Latombe J-C, Lin MC (eds) *Algorithmic foundations of robotics IX*. Springer, Berlin Heidelberg, pp 19–35
34. Fox D, Burgard W, Thrun S (1997 March) The dynamic window approach to collision avoidance. *IEEE Robot Autom Mag* 4(1):23–33
35. Simmons R (1996 April) The curvature-velocity method for local obstacle avoidance. In *Proceedings of IEEE International Conference on Robotics and Automation*. 4:3375–3382
36. Nak Yong K, Simmons RG (1998 October) The lane-curvature method for local obstacle avoidance. In: *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98): Innovations in Theory, Practice and Applications* 3:1615–1621 <https://doi.org/10.1109/IROS.1998.724829>
37. Benayas JA, Fernández JL, Sanz R, Diéguez AR (2002 January) The beam-curvature method: a new approach for improving local realtime obstacle avoidance. *IFAC Proceed-Ing*s 35(1):409–414
38. Reif J, Wang H (1998) The complexity of the two dimensional curvature-constrained shortest-path problem. In *Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective*, A. K. Peters, Ltd. (event-place, Natick, MA, USA: Houston, Texas, USA), 49–57
39. Moustiris GP, Tzafestas CS (2016 Dec) Intention-based front-following control for an intelligent robotic rollator in indoor environments. In 2016 ieeee symposium series on computational intelligence (ssci), pp 1–7
40. Graham JE, Fisher SR, Bergés I-M, Kuo Y-F, Ostir GV (2010 11) Walking speed threshold for classifying walking independence in hospitalized older adults. *Phys Ther* 90(11):1591–1597. Retrieved from <https://academic.oup.com/ptj/articlepdf/90/11/1591/31674633/ptj1591.pdf>
41. Thrun S, Burgard W, Fox D (2005) *Probabilistic robotics*. The MIT Press
42. Székely GJ, Rizzo ML (2013) Energy statistics: a class of statistics based on distances. *J Stat Plann Inference* 143(8):1249–1272

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

George Moustiris holds a M.Eng. in Electrical & Computer Engineering from ECE/AUTH (2003) and a Ph.D. in robotics and control from ECE/NTUA (2010). Since 2011, he has been a Research Associate at the Intelligent Robotics and Automation Laboratory (IRAL), ECE/NTUA, conducting research in artificial intelligence, robotics, HRI, haptics, and automatic control. He currently serves as a Senior Researcher at the Institute of Robotics (IRO), ATHENA Research Center, where he acts as Principal Investigator in healthcare and field robotics research projects.

Costas Tzafestas is a Professor of Robotics at the School of Electrical and Computer Engineering (ECE) of the National Technical University of Athens (NTUA). He holds an Electrical and Computer Engineering Degree from NTUA (1993), as well as a D.E.A. (1994) and Ph.D. (1998) in Robotics from the Université Pierre et Marie Curie (UPMC, Paris 6, France). Before joining NTUA in 2003, he served as a Research Associate at the Institute of Informatics and Telecommunications of the National Center for Scientific Research “Demokritos” (Greece). His main research interests include cognitive assistive robotics, human-robot interaction, haptics and telerobotics, also spanning

intelligent control and robot learning, with applications in healthcare robotics, advanced manipulation and mobile robotics. He has authored or co-authored more than 180 scientific publications and has participated as Project Coordinator, PI, or Scientific Manager in several European and national research projects in the above fields. He currently serves as an Executive Board Editor of the Journal of Intelligent and Robotic Systems. He is also a member of the Executive Committee of the Hellenic Robotics Center of Excellence (HERON), funded through the EU Teaming for Excellence programme.