2021 I-RIM Conference
October 8-10, Rome, Italy
ISBN: 9788894580525
DOI:10.5281/zenodo.5900505

A Novel Framework for Multi-Agent Navigation in Human-Shared Environments

Manuel Boldrer, Daniele Fontanelli
Dept. of Industrial Engineering
University of Trento
Trento, Italy

Alessandro Antonucci, Luigi Palopoli
Dept. of Information Engineering and Computer Science
University of Trento
Trento, Italy

***Abstract*—We propose a hierarchical framework that combines global path planning, local path planning and reactive strategies, ensuring a safe and socially-aware navigation. We test the effectiveness our approach on real robotic platforms.**

***Index Terms*—Multi-robot navigation, Socially-aware navigation, Distributed systems.**

I. Introduction

The ability to move in dynamic and partially unknown environments is a fundamental requirement for modern ground robots, either considering legged or wheeled robots. When we deal with robots navigation in human-shared environments, it turns out to be necessary meet multiple requirements:

R1: Robustness and safety in navigation. The first essential requirement concerns the safety guarantees. In particular, we have to ensure collision avoidance with obstacles, other robots and human beings. This condition must be ensured even in the face of inaccurate obstacle's velocity estimation or inaccurate robot localisation.

R2: Socially-aware motion planning. Ensuring safety is not enough in a human-shared environment. Indeed the robot has to stay clear enough from the human private space, it has to follow smooth trajectories and it should anticipate the human motion intentions as well. In this way, the robot presence will not intimidate the humans in the scene, since the robot follows trajectories that are easily predictable by the pedestrian.

R3: Multi-agent coordination. Most often, we are considering spaces that are shared with multiple robots, thus requiring the definition of a distributed coordination strategy between the robots.

R4: Dynamic environment. The prior knowledge of the mission space is an obvious prerequisite: however, the navigation algorithm should be able to 1. quickly react to unmapped obstacles that are detected by on-board sensors and 2. drive the robot towards its goal location.

R5: Computation efficiency. The last requirement regards the computation efficiency. This is necessary mainly for two reasons: 1. the robot should be able to compute the control inputs with high frequency to ensure a timely response to unexpected situations and 2. we need to adopt lean hardware with reduced costs and low energy consumption.

Related works To the best of the Authors' knowledge, a comprehensive solution that satisfies all the requirements mentioned above is not available in the literature.

In [1] the authors provide a brief survey on robots navigation in human environments; they identify mainly three different approaches to the problem: *reactive methods*, *predictive planners* and *learning-based methods*. In the following we underline the pros and cons of the three methods just mentioned.

Reactive methods: These methods synthesise directly the control inputs on the basis of local information. The most popular examples are Force field based [2] or Velocity Obstacle (VO) based methods [3]. These methods are by construction computationally efficient (**R5**) and typically they can be extended to the multi-agent case (**R3**). As the name suggests, these methods are designed to react promptly and safely to the presence of obstacles (**R1**). Nevertheless, the majority of these approaches, e.g. VO-based methods, need a very good accuracy in the obstacles' velocity estimation; when this is not possible, collisions cannot be ruled out (failing **R1**). Moreover, the main drawback is induced by the short-sighted decision mechanism, which leads to 1. possible deadlocking conditions (failing **R4**), and 2. paths that may not stay clear enough from the pedestrian private space, hence violating the socially-aware requirement (failing **R2**).

Predictive methods: The predictive methods synthesise a safe path, which optimise a desired cost function, that takes into account local information and also a reliable human motion prediction. As long as the human motion prediction perfectly fits the trajectory that the person will go through and the cost function is properly selected, the planned path results to be safe (**R1**) and socially-aware (**R2**). By relying on these methods, deadlock is typically avoided because of the longer planning horizon (**R4**). On the other hand these methods are computationally expensive, hence they cannot execute with a sufficiently high frequency (failing **R5**). As a consequence, if the human motion prediction results to be inaccurate, safety cannot be ensured (failing **R1**). Moreover, the multi-agent coordination cannot be straightforwardly integrated; some adaptation is possible but priorities among agents have to be assigned (failing **R3**).

Learning-based methods: Finally, learning-based methods can obtain a good level of safety and socially-awareness (**R1, R2**) by training the robots with a lot of examples. However, these techniques may easily fall in what is called data overfitting, which leads to the incapability to generalise the behaviour in different or dynamic environments (failing

R4). Rigorous safety guarantees cannot be provided by relying on pure learning-based methods (failing **R1**), moreover multi-agent management (**R3**) increases considerably the degree of complexity at the learning stage. Nonetheless, the computational requirements are usually acceptable for real-time navigation aims (**R5**).

II. Solution overview

Our proposal [4] identify a hierarchical architecture composed by 3 layers that works together in order to satisfy the requirements **R1 - R5**. As we depict in Figure 1 we called the three layers: *Global path planner* (GPP), *Local path planner + Human motion prediction* (LPP) and *Lloyd-based controller* (LB) . Thanks to the interaction between the three layers we achieve: guarantees on safety and robustness thanks to the imposed safety redundancy and the high frequency correction implemented by LB **(R1)**; generation of socially-aware trajectories thanks to the LPP module, which generates smooth and safe paths taking into account human motion predictions to anticipate their intentions and to stay clear enough to their private space **(R2)**; an effective distributed coordination for the multiple robots in the scene, due to the LB design **(R3)**; a successful management of partially dynamic environments and the progress towards the goal location because of the interaction between GPP and LPP layers **(R4)**; quick computation of the control inputs thanks to the LB controller and a moderate computational power requirement to run the overall algorithm **(R5)**.

As it can be noticed in Figure 1, the inputs for the i–th agent are the CAD map, the initial position $p_i(0)$ and the final goal position e_i. The sensors used for the localisation of the unicycle-like robot adopted in the experiments are the wheel encoders, the Inertial Measurements unit (IMU) and the Light Detection and Ranging (LiDAR). The GPP layer uses the CAD map, $p_i(0)$ and e_i to generate the global path $\mathcal{P}_{g,i}$, which safely links the initial robot position $p_i(0)$ to its final goal location e_i. The global path is computed only upon request, typically at the beginning of the mission. The global path $\mathcal{P}_{g,i}$ is passed to the second layer, i.e. the LPP, to generate the local path $\mathcal{P}_{l,i}(t)$ incorporating the human beings motion predictions $\mathcal{P}_h(t, t+n_f\delta_t)$. This layer uses local information in a range of $5-10$ m and it is computed with a frequency of $1.25-2$ Hz. The path $\mathcal{P}_{l,i}(t)$ generated fits the requirements (**R1**), (**R2**) and (**R4**), without considering the presence of the other robots. Then, the local path $\mathcal{P}_{l,i}(t)$ is passed to the LB controller that generates the forward velocity v_i and the angular velocity ω_i of the unicycle robot. The LB layer has a reduced space of interest i.e. 3 m, and it generates updated control inputs every 50 ms. The communication network is mainly used to share information about the neighbours positions $p_j(t)$, to coordinate the robots, and the data collected from the environment. Roughly speaking, the LB controller builds a safe region around the robot and it checks if the $\mathcal{P}_{l,i}$ path is safe: if so, the robot basically follows the $\mathcal{P}_{l,i}$ path; otherwise it computes a safety-preserving deviation. The $\mathcal{P}_{l,i}$ path may be unsafe mainly for two reasons 1. there is another

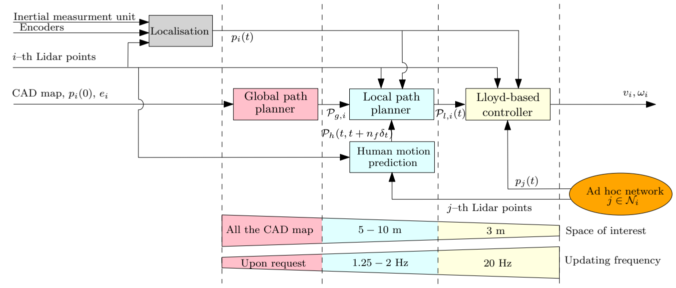

Fig. 1. Proposed hierarchical architecture for the i–th agent.

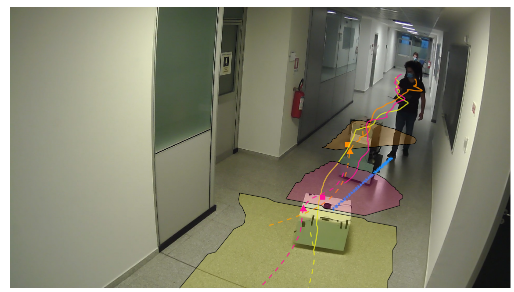

Fig. 2. Navigation with 3 robots in human shared environment. We depict the past trajectories (solid lines), the cells computed by LB for each robot, the planned paths by using LPP (dashed lines) and the human motion predictions (lines with asterisks).

robot on the planned path (LPP does not consider the presence of other robots) and 2. the environment is highly dynamic, hence the updating frequency may be too slow.

We tested the proposed framework on real robotic platforms fully developed at the University of Trento and we proved through extensive experiments the effectiveness of our approach. In particular, Figure 2 shows a snapshot for an experiment conducted with 3 robots in a human shared environment, where the paths and the attention space of the LB are clearly visible.

III. Future works

In the near future we plan to integrate more sensors with the aim of obtaining more accurate human motion predictions and hence improve the LPP. Other studies should be done to integrate in the LPP a distributed and computationally efficient (not sequential) multi-agent management system.

References

[1] J. Cheng, H. Cheng, M. Q.-H. Meng, and H. Zhang, “Autonomous navigation by mobile robots in human environments: A survey,” in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 1981–1986.

[2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[3] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[4] M. Boldrer, A. Antonucci, P. Bevilacqua, L. Palopoli, and D. Fontanelli, “Multi-agent navigation in human-shared environments: a safe and socially-aware approach,” *Robotics and Autonomous Systems*, submitted.