Experiments of Obstacles and Collision Avoidance with a Distributed Multi-Robot System

Filippo Arrichiello and Stefano Chiaverini
Department of Electrical and Information Engineering
University of Cassino and Southern Lazio
03043 Cassino, Italy
{f.arrichiello, chiaverini}@unicas.it

Vaibhav Kumar Mehta
L.N. Mittal Institute of Information Technology
Jaipur, India
vaibhavmehta@ieee.org

Abstract—In this paper we describe preliminary experiments of obstacle and collision avoidance with a distributed multi-robot system composed of grounded mobile robots equipped with laser range finder. We implemented a localization algorithm to make the robots avail autonomous navigation in indoor environments, and then we apply a behavior based technique, namely the Null-Space based Behavioral control, to make the robots navigate avoiding collision among themselves and with obstacles detected with the laser. Several tests have been performed using a team of up to four Khepera III mobile robots equipped with Hokuyo URG laser range finder and communicating via ad-hoc wireless network.

I. INTRODUCTION

Multi-robot systems received an increasing attention in the last decades due to the advantages they present w.r.t. single robot in terms of fault tolerance, flexibility in mission execution, possibility to use distributed sensing and actuation, etc. The work [12] presents an overview on multi-robot systems focusing on aspects like system architecture, communication, task allocation, and application; while the work [11] deals with the concepts of distributed intelligence and introduces a classification based on exhibited interactions.

Recent researches in this field are mainly focused on distributed and cooperative robotic systems [4], i.e. systems where each robot interacts only with its neighbors and, together, they generate the global behavior of the team. The work [8] presents an overview on networked robots, that is multiple robots that cooperate by network communication, and explores the research challenges due to the interaction among control, communication and perception. The book [9] provides an introduction to analysis and design of dynamic multi agent networks focusing on graph theoretic methods.

Several recent studies deal with the development of distributed control approaches for multi-robot systems with the aim of achieving a global task (e.g., controlling the geometrical centroid) by using distributed controllers. An aspect of the mentioned research poses the challenging problem of consensus for multi-agent systems, i.e., reaching an agreement regarding a certain variable depending on the state of all the agents. In references [13] and [10] the consensus algorithms are investigated with emphasis on robustness, time-delays and performance bounds; while the results in [6] are related to the stability analysis of several decentralized strategies that achieve an emergent behavior. Networked robots give rise also to the problem of connectivity maintenance, as addressed in [7], that is the problem of how to ensure that the group of robots stay connected while accomplishing their task.

With the final aim of developing and testing distributed control algorithms, we build up a completely distribute experimental set-up composed of several Khepera III mobile robots equipped with Laser Range Finder (LRF) and communicating via ad-hoc wireless network. Here we present preliminary experiments we performed with such platform. In particular, we at first implemented on board each robot a localization algorithm taken from the literature [14] that allows the robots to navigate in indoor environment (structured environment and corridor). Specifically, the localization algorithm is a feature-based Extended Kalman Filter (EKF) that uses odometry measurement for motion prediction and laser information for measurement correction. Then, to control the robots motion, we refer to the Null-Space based Behavioral control, developed by some of the Authors of this paper [2]-[3], and we use it to make the robots navigate towards waypoints, exchanging information via ad-hoc network, and avoiding collision among themselves and with dynamic obstacles detected on their path. Results of tests with a single robot avoiding unpredicted obstacles and with four robots avoiding collisions are presented.
II. EXPERIMENTAL PLATFORM

For the experiments presented here, we used a team of four Khepera III mobile robots, produced by K-Team corporation, that are small size (12cm diameter) differential drive mobile robots whose base is equipped with several proximity sensors (11 infrared sensors TCRT5000 Vishay Telefunkenand, 5 ultrasound sensors 400ST/R100 Midas Components Ltd). The robots have been equipped with an extension board, namely the Korebot II, that is an embedded platform based on the gumstix Verdex PRO with a Marvell PXA270 XScale processor @ 600MHz, with 128MB RAM and 32MB Flash memories. The Korebot II provides a standard embedded Linux operating system (Angstrom distribution, kernel 2.6) and, to program it, we made use of the Khepera III toolbox developed from the Distributed Intelligent Systems and Algorithms Laboratory (DISAL) of EPFL.

To allow robot-robot and, eventually, robot-pc communication, the robots have been equipped with an IEEE 802.11 wireless card. In particular, the robots communicate through a wireless ad-hoc network built with the idea to perform completely distributed experiments. In fact, the final objective is to perform experiments in (indoor) large environments where the robots directly communicate only with their neighbors (i.e., other robots within the communication range) avoiding fixed access points. Apart from the configuration of ad-hoc network parameters, the communication strategy implemented on board each robot exercises a multi-process architecture where each robot is able to broadcast information using UDP/IP protocol, and store information received from its neighbors. Eventually, if a robot needs to send specific information to a not-directly connected robot, it can use multi-hop communication. We have tested specific linux functions as ip forwarding and olsrd daemon for the same.

To improve the robots perception capabilities we equipped them with Hokuyo LRF URG-04LX-UG01. The LRF is connected to the Korebot II through USB but we provide it power supply from external batteries. The Hokuyo LRF has adequate performance in terms of weight, power consumption, range and accuracy, to properly fit with Khepera III robots and allow them navigating in an indoor environment.

III. EXTENDED KALMAN FILTER LOCALIZATION

Mobile robot localization (often called as position estimation or position tracking) is the problem of determining the pose of a robot relative to a given map of the environment, and it is one of the most basic perceptual problem in robotics. The maps are usually described in a global coordinate system, thus, localization is the process of establishing correspondence between the map coordinate system and the robots local coordinate system. Unfortunately, for indoor or GPS denied environment, pose usually can not be directly sensed, thus it has to be inferred from data. However, a single measurement is usually not sufficient to properly determine the pose, and the robot has to integrate data over time. Thus, for the localization, the robot needs both a-priory information about the environment and information from exteroceptive sensors during the navigation.

In this paper we refer to the approach presented in [14]-[5] and references therein; specifically we implemented a feature based Extended Kalman Filter (EKF) localization algorithm where the reference features are represented as line segments, and data association and measurement correction are performed in the Hough domain. Lines, as geometric feature, provide strong and accurate information, but with far less numbers than that of points. This solution properly fits with our specifications since it works well for indoor environment, where the number of linear features is high (walls, doors, closets), it does not require high computation (which is of primary importance for the robotic platform we are using), it is robust to unpredicted obstacles and fit well with our obstacle avoidance strategy.

In particular, robot localization is performed referring to a discrete-time formulation of the EKF that, as well known, is an approximate estimator for the states and parameters of nonlinear systems. Here, we assume that the state and the measurement dynamics are respectively governed by:

\[
\begin{align*}
\mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \\
\mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k
\end{align*}
\]

where \(\mathbf{x}_k\) is the state (pose and orientation), \(\mathbf{u}_k\) is the control input, \(\mathbf{y}_k\) is the measurements, \(\mathbf{w}_k\) and \(\mathbf{v}_k\) are respectively process and measurement zero mean Gaussian noise with \(R_w\) and \(R_v\) covariance matrixes. Moreover we assume that we can measure the true initial location \(\mathbf{x}_0\) of the robot with some precision and we can use it to initialize the EKF with some uncertainty. Then, the EKF implementation follows the two-step iterations:

A. Prediction of state and estimation of error covariance

The state prediction is performed on the base of the knowledge of kinematic model parameters and input values. In the specific case, it is performed using the input values \(\mathbf{u}_k = [s_l \ s_r]^T\) (i.e. the left and right wheels motion displacements, see fig. 2) and the knowledge of the odometric model parameters. The process covariance is \(R_w = \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}\) where we suppose a linear growth of the error model covariance \(\sigma_l = k_l |s_l|, \sigma_r = k_r |s_r|\). Thus, the system state is given by:

\[
\mathbf{x}_k = [x_k \ y_k \ \theta_k]^T
\]

and the state prediction update \(\tilde{\mathbf{x}}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)\) is given by:

\[
\begin{bmatrix}
\tilde{x}_{k+1} \\
\tilde{y}_{k+1} \\
\tilde{\theta}_{k+1}
\end{bmatrix} = 
\begin{bmatrix}
\frac{s_l + s_r}{2} \\
\frac{s_l + s_r}{2} \\
\frac{s_l + s_r}{2}
\end{bmatrix} \begin{bmatrix}
\cos(\theta_k) + \frac{s_l - s_r}{2b} \\
\sin(\theta_k) + \frac{s_r - s_l}{2b} \\
\frac{s_r - s_l}{b}
\end{bmatrix}
\]

where \(k_l, k_r\) are the wheel kinematic parameters.
where \( b \) is the distance between the wheels. The estimation error covariance is given by

\[
P_{k+1}^- = F_k P_k^+ F_k^T + G_k R_k G_k^T
\]

where \( F_k = \frac{\partial f}{\partial x} x=x^k, u=\bar{u}_k \) and \( G_k = \frac{\partial f}{\partial u} x=x^k, u=\bar{u}_k \) that, for the specific case are:

\[
F_k = \begin{bmatrix}
1 & 0 & -\frac{(s_{k+1} + s_{k})}{2} \sin(\theta_{k+1} + \frac{s_{k+1} - s_{k}}{2b}) \\
0 & 1 & \frac{(s_{k+1} + s_{k})}{2} \cos(\theta_{k+1} + \frac{s_{k+1} - s_{k}}{2b}) \\
0 & 0 & 1
\end{bmatrix}
\]

\[
G_k = \begin{bmatrix}
\frac{1}{2} (\cos(\theta) + S \sin(\theta)) & \frac{1}{2} (\cos(\theta) - S \sin(\theta)) & \frac{1}{2} (\sin(\theta) + S \cos(\theta)) \\
\frac{1}{2} (\sin(\theta) - S \cos(\theta)) & \frac{1}{2} (\sin(\theta) + S \cos(\theta)) & \frac{1}{2} \sin(\theta)
\end{bmatrix}
\]

where \( \theta = \theta_{k+1} + \frac{s_{k+1} - s_{k}}{2b} \) and \( S = \frac{(s_{k+1} + s_{k})}{2b} \).

The measurement update of state and estimation error

The measurement update is performed using line information from the LRF. Specifically, to extract line segments from the set of points of the LRF scan (see fig. 3), we use recursive line splitting method[1], which is reliable and fast. Various steps of this method are described as follows:

1) form a line from start and end points of the scan;
2) find the point with the biggest distance to this line;
3) if its distance is lower than a certain threshold, a line segment is found;
4) if not, split the point group at this point and repeat.

After getting the line segments information, they were represented in the Hough Domain. The Hough transform allows to represent a line represented in a Cartesian coordinate using two parameters \( \rho \) and \( \alpha \) that respectively represent the distance of the origin from the line and the angle of the segments starting from the origin and perpendicular to the line (see fig. 4). Thus, for each segments, we store \( \rho \) and \( \alpha \) and the coordinates of the vertex points.

Obtained the list of observed features, we have to perform the Data Association, that is, try to associate each observed feature with a corresponding feature of the known map. To allow the correspondence, both maps (a-priori and observed) were expressed in the same reference system (the robot local reference system), and the comparisons and the association were made in the Hough domain following the steps:

1) For all the observed landmark (NO), store in a list A Hough coordinates and segments vertex coordinates;
2) For each observed landmark in A find, if there exists, the features of the map with a certain threshold, i.e. the points in Hough domain with distance from the landmark lower than a certain threshold; store them in a list B (see fig. 5);
3) Among the different map features in the list B, find the segment having mid point with the minimum distance from the observed segment midpoint;
4) If the distance between the midpoints is lower than a certain threshold, then associate this feature to the observed landmark.

Indicating the \textit{NO} observed landmarks with successful association procedure as:

\[
y_k = [r_1 \psi_1][r_2 \psi_2] \cdots [r_{NO} \psi_{NO}]^T
\]
and indicating the associated map landmarks as:

\[ h(\tilde{x}_{k+1}^-) = [\rho_1 \alpha_1] \cdots [\rho_{NO} \alpha_{NO}]^T \]

then the measurement updates of state and estimation error of the EKF has the form:

\[ K_{k+1} = P_{k+1}^- H_k^T [H_k P_{k+1}^- H_k^T + R_v]^{-1} \]

\[ \hat{y}_{k+1} = y_k - h(\tilde{x}_{k+1}^-) \]

\[ \tilde{x}_{k+1}^+ = \tilde{x}_{k+1}^- + K_{k+1} \hat{y}_{k+1} \]

\[ P_{k+1}^+ = (I - K_{k+1} H_k) P_{k+1}^- \]

where \( H_k = \partial h(\cdot)|_{x=\tilde{x}_{k+1}^-} = [H_{1,k}^T \cdots H_{NO,k}^T]^T \) and

\[ H_{i,k} = \begin{bmatrix} -\cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \]

IV. NULL-SPACE BASED BEHAVIORAL CONTROL

To manage robots’ motions we refer to the Null-Space based Behavioral (NSB) control, that is a behavior based approach developed by some of the Authors to control both single and multiple robots (see e.g. [2]-[3]). This control approach allows to simultaneously manage multiple tasks arranged in priority. The basic idea is that the lower priority tasks are solved in the null-space of higher priority ones so that they do not effect them. We report the basic concepts in the following.

Let define the position of the \( j \)th robot as \( p_j = [x_j \ y_j]^T \) and the generic task variable to be controlled as \( \sigma \in \mathbb{R}^m \)

\[ \sigma = f(p_j) \]

The corresponding differential relationship is:

\[ \sigma = \frac{\partial f(p_j)}{\partial p_j} v_j = J(p_j) v_j \]

where \( J \in \mathbb{R}^{m \times 2} \) is the robot configuration-dependent task Jacobian matrix and \( v_j \in \mathbb{R}^2 \) is the robot velocity.

The motion directives to the robot, i.e. velocity reference command, are elaborated as:

\[ v_{d,j} = J^T(\dot{\sigma}_d + \Lambda \dot{\sigma}) \]

where \( J^T(J J^T)^{-1}, \Lambda \) is a suitable constant positive-definite matrix of gains and \( \dot{\sigma} = \sigma_d - \sigma \) is the task error.

Let us consider the mission for the \( j \)th robot composed by two elementary tasks. Using the subscript \( i \) referring to the \( i \)th task quantities for the \( j \)th robot, on the analogy of the above equation, the \( i \)th task velocity is computed as

\[ v_i = J_i^T(\dot{\sigma}_{i,d} + \Lambda_i \dot{\sigma}_i) \]

If the subscript \( i \) also denotes the degree of priority of the task with, e.g., Task 1 being the highest-priority one, according to [3] the final motion command to the robot is modified into:

\[ v_{f,j} = v_1 + (I - J_1^T J_1) v_2 = v_1 + N_1 v_2 \]

where \( I \) is the identity matrix of proper dimensions and \( N_i \) is the null-space projection matrix of the \( i \)th task. Remarkably, above eq. has a nice geometrical interpretation. Each task velocity is computed as if it were acting alone; then, before adding its contribution to the vehicle velocity, a lower-priority task is projected onto the null space of the immediately higher priority task so as to remove those velocity components that would conflict with it.

For the experiments presented in the next session we considered a multi-robot system where each robot accomplishes an individual task of moving toward assigned waypoints (or goals) avoiding obstacles and collision with other vehicles. Thus, the mission of each robot is decomposed in two elementary tasks:

- **Move to Goal.** The move-to-goal task function is:

\[ \sigma_g = p \in \mathbb{R}^2 \]

while the desired values is \( \sigma_{g,d} = p_g \) where \( p_g \) are the coordinates of the goal. Since \( J_g = J_g^T = I \in \mathbb{R}^{2 \times 2} \), the output velocity is given by

\[ v_g = A_g(p_g - p) \]

that is a velocity in the goal direction proportional to the distance from the goal \( p_g \).

- **Obstacles and collision avoidance.** In presence of an obstacle in \( p_o \) in the advancing direction, the aim of the task is to keep the robot at a safe distance from it. Therefore the task function is

\[ \sigma_o = ||p - p_o|| \in \mathbb{R}, \]

and \( \sigma_{o,d} = S_d \) where \( S_d \) is the safety distance from the obstacle. Then, it holds

\[ J_o = \hat{v}^T \in \mathbb{R}^{1 \times 2} \]

where \( \hat{v} = \frac{p - p_o}{||p - p_o||} \) is the unit vector aligned with the obstacle-to-vehicle direction. Therefore, the task output is a velocity, in the robot-obstacle direction, that keeps the robot at a safe distance from the obstacle:

\[ v_o = J_o^T \lambda_o (S_d - ||p - p_o||). \]

while the task null-space matrix \( I - J_o^T J_o \) projects velocity vectors of lower priority task in the direction tangent to a circle centered in \( p_o \) and passing for \( p \).

In presence of a linear obstacle, the task function is analogous to the previous case with the difference that \( p_o \) is the point of the segment closest to \( p \).

The obstacle-avoidance is usually the highest priority task since its achievement is of crucial importance to preserve the integrity of the vehicle, however it is activated only in specific conditions, that is when the robot is moving toward the obstacle and when the distance is lower than a certain threshold (see fig. 6). In this case, the implementation of
the NSB approach makes the robot moving toward the goal sliding around the (point or linear) obstacle keeping a certain safety distance.

It is worth noticing that the output of the NSB is a linear velocity command elaborated considering the robot as a material point. To use the NSB to control a non-holonomic vehicle as the Khepera III, the output velocity of the NSB is passed to a Low-Level control that properly elaborates angular and advancing velocity and, using odometric model parameters, converts it to left/right wheels velocity command.

In the implementation of the algorithms, a saturation is performed on the resulting velocity in order to feed the system with limited amplitude signals.

V. EXPERIMENTAL RESULTS

In this section we will illustrate two experiments, the first done with a single robot localizing using the EKF algorithm and avoiding obstacles using the NSB control, and the second performed with four robots avoiding collisions using both LRF and ad-hoc wireless communication.

In the first experiment the robot was commanded to iteratively move between two waypoints whose coordinate are assigned relatively to the global map of the environment. The robot had two tasks in hand: move towards the goal and avoid dynamic obstacles eventually detected in the environment. The obstacle avoidance, when active, was the highest priority task. Figure 7 shows a single snapshot of an iteration where the robot detects and avoids an obstacle not present in the environment map. Figures 8 shows several snapshots illustrating a reconstruction from experimental data of the robot behavior. In particular, the blu perimeter represents the map of the environment available to the robot, while the black segments are the segments extracted from the LRF data; it is worth noticing that some of that segments are associated to the map and allow the localization, while other segments are used for obstacle avoidance. The red triangle represents the differential drive robot with the green vector representing the \( v_g \) and the blue vector representing the \( v_f \) velocities for the robot. The blue/magenta circles represent the reference values for the obstacle avoidance task, respectively threshold for detection of an obstacle and safety distance \( S_d \). Figure 8 shows the sliding behavior of the robot around the black colored obstacle detected from the LRF. The experiment was done with the robot’s saturation velocity set to 20 cm/s, \( \Lambda_g \) set to a diagonal matrix with elements 0.8, and \( \lambda_o \) set to 1. A complete video of the experiment that shows the robustness of the localization and obstacle avoidance algorithms even in the case of a human entering the arena is available at http://webuser.unicas.it/lai/robotica/video.html

To test the robustness of the localization algorithm we also perform several tests in noisy indoor environment (a corridor with open doors, closets with glass wall, people walking around); results are not reported here for space limitation but reconstruction form experimental data is available at the previous web-page.

The second experiment was performed using four Khepera III robots iteratively moving among waypoints. The robots were initially displaced in a cross configuration where each robot
has to exchange its position with his diametral opposite robot. The robots are not coordinate but they exchange their position information using wireless ad-hoc network to facilitate the obstacle avoidance. As for the previous experiment, the saturation velocity is set to 20 cm/s, $A_g$ set to 0.8$I_2$ and $\lambda_0$ set to 1. Figure 9 shows a single snapshot of the experiment, while Fig. 10 shows the reconstruction from the experimental data taken from one of the robot. As shown in the video available at the previous web-page, the robots also avoid collisions with unpredicted obstacle.

VI. CONCLUSION

In this paper we presents preliminary experiments we performed with a distributed multi-robot system composed of four Khepera III mobile robots with LRF. We implemented and tested a feature-based EKF localization algorithm and we used it together with a behavior-based motion control, the NSB, to allow the robots navigating in indoor environment avoiding unpredicted obstacles and collisions among themselves. Despite in the performed experiments the cooperation among robots is limited to the position information exchange for obstacle avoidance, testing the robustness and the performance of the system represented a preliminary result that allows to focus our future work on the development and testing of distributed coordination control algorithm for multi-robot systems.

ACKNOWLEDGMENTS

The Authors would like to thank prof. Gianluca Antonelli for his valuable comments and suggestions for the realization of this work. The research leading to these results has received funding from the Italian Government, under Grant FIRB - Futuro in ricerca 2008 n. RBFR08QWUV (NECTAR project), and under Grant PRIN 2008 n. 2008EPKCHX (MEMONET project).

REFERENCES