Cooperative navigation using environment compliant robot formations

P. Urcola, L. Riazuelo, M. T. Lázaro and L. Montano
Instituto de Investigación en Ingeniería de Aragón, University of Zaragoza (Spain)
Email: {urcola, riazuelo, 512971, montano} @unizar.es

Abstract—This paper reports an autonomous cooperative navigation system for robot formations in realistic scenarios. The formation movement control is based on a virtual structure composed by spring-damper elements, which allows the formation to comply with the environment shape. A different navigation strategy is applied to the leader of the formation and to the rest of robots of the team. The leader plans the trajectories by using a two-level path planner with obstacle avoidance capabilities. The motion of the follower robots is controlled by the virtual structure, which adapts to the environment while the leader is tracked, taking into account the kinodynamic constraints of the vehicles. The system is evaluated in experiments carried out in simulations and with real platforms, some of them made in realistic and complex urban scenarios.

I. INTRODUCTION

In the last years, many efforts have been made in researching and developing applications of mobile multi-robot systems. One of the main issues of these works is the maintenance of some kind of group formation, more or less adaptable to the environment, while moving the team to a goal. This topic is one key point in some real outdoors applications, where robots move through big scenarios and cooperation between them is needed. Unknown zones exploration, surveillance, connectivity maintenance or guiding are examples of applications in which cooperating robot teams are a good solution.

Concretely, in URUS project [1] an evacuation in urban environment mission is defined. When a fire is detected, robots guide confused people through safe ways to a safe area. While they are moving, they are blocking dangerous zones to avoid people to enter on them.

For all these kind of tasks, a robot team needs some flexibility to fit the environment as they are navigating and finding different obstacles such as narrow streets, intersections or curves.

There are many papers considering formation maintenance without obstacle avoidance. In [2], a virtual structure is defined to model the shape of the formation. They employ graphs to model this structure where robots are placed in the vertices and links represent the relative positions between them. This model is quite rigid because it does not consider neither changes in formation topology nor flexible edges.

Many works tried to model formation control using physic analogies, because it is easier to analyse the behaviour using mathematical methods like Lyapunov functions [3] to study system stability. Concretely, virtual potential fields are applied to model the influence of the location of each robot in the movement of the others. In [4], potentials are used to allow formation flocking and schooling manoeuvres of very populated groups of robots.

Derived from potential fields we can consider virtual springs systems. These kind of approximations use the Hook’s Law to compute forces between robots, giving some flexibility to the structure and then, smoothness to the movement. Works like [5], [6] and recently [7] use springs and dampers to model relative control among robots inside the formation. We use in this work a similar structure to control the formation movement.

Works cited before do not consider the environment influence in formation movement.

One approximation for obstacle avoidance in formation is to compute a configuration space for the whole formation as explained in [8]. This approach is not adequate for our purposes because a complete map is needed and we pretend to work in unknown scenarios too. Another solution proposed in [9] for providing flexibility is changing the interconnections between robots to modify the shape of the formation while moving. The problem of this solution resides in how to decide the best formation shape depending on the environment.

In [10], potential fields are utilised to fuse formation control and obstacle avoidance techniques. This work and [11] are the closest ones to our proposal. They both use the idea of forces between robots to keep formation and forces from the environment to avoid obstacles, but neither robot movement constraints nor complex and realistic environments are considered.

We propose here the control of the movement of robot formations by considering both kinematic and dynamic constraints for the robots and navigation in realistic scenarios with obstacles, where formations have to comply to the environment shape, while maintaining approximately the formation topology.

Our proposal also takes into account communication issues. We use the real time protocol over wireless ad-hoc networks defined in [12] for data interchange in cooperative control. This ensures that all the data shared by the robots are available for every component in the team.

In sections III and IV, we present the formation control schema and the environment modelling for obstacle avoid-
 ance. In section V we show how to fuse formation control and obstacle avoidance with path planning. To conclude, we present in sections VI and VII the results obtained in simulated and real experiments and the conclusions we extract from this work.

II. SYSTEM OVERVIEW

The formation has only one leader. This robot plans the trajectory to the goal and tracks this path guiding the others. Path planning and obstacle avoidance algorithms are applied to complete this task. The rest of team members follow the leader controlled by a model based on a virtual structure.

Fig. 1 represents the general architecture for movement control. Each block represents a module that is running inside every follower robot.

Robots gather data from their on board sensors and, depending on the type of sensor, information is used for different purposes. Environment perception and analysis block takes as input laser scans data. Localization module integrates odometry and gyroscope data from the robot to estimate the position of the robot. To improve this estimation, we use scan matching techniques for indoor experiments and GPS integration for outdoor environments. The environment analysis module calculate the influence of obstacles on the robot computed as a force (see section IV). This influence and the goal attraction produce the external forces that will affect the robot movement. Formation structure forces are generated by virtual springs and dampers (see section III) that model the relative localization between robots. These forces are computed in the cooperative formation control module.

Once an unique force is computed for each follower, it is used as an input to the motion generator module. It computes velocity commands for the robots considering kinematic and dynamic constraints. As we are using non-holonomic robots, this command consists in linear and rotational velocities to send to the robot controller.

This paper is focused on integrating environment information on the formation control to adapt navigation to the scenario.

III. FORMATION STRUCTURE

A. Structure model

In order to accomplish a robot team movement in formation is necessary to fix the structure the robots have to maintain. In this section we present the model of virtual structure that we use. This model is based on a spring-damper analogy. In order to incorporate obstacle avoidance capabilities it is necessary to build a structure that can be deformed in such a way the robots can avoid the obstacles while maintain the initial formation topology. Fig. 2 depicts the virtual structure proposed. In it the robots are linked with linear spring-damper components, torsional springs or both of them. With the linear spring-damper link we achieve to maintain the distance between the robots and the torsional springs force the robots to maintain a given angle between them. This model provides the desired behaviour to the formation navigation.

The model designed allows to change the links between robots in order to adopt different initial structures.

The main force is $G_L$, applied to the robot leader (L). This is the virtual force exercised by the goal on the leader which attracts the robot to it. A first approximation of this force is computed as a function of the given leader’s maximum desired velocity and so it is limited. In section V we present the path planning integration and how this force $G_L$ is computed according to the plan.

The spring-damper link between the robots generates a force $SD_i$. This force is defined as:

$$SD_i = \sum_{j=1}^{N} sd_{ij}a_{ij} + \sum_{j=1}^{N} st_{ij}b_{ij}$$

(1)

where the elements $a_{ij}$ of the matrix $A$ represent the linear spring-damper links between robots, and the elements $b_{ij}$ of the matrix $B$ represent the torsional spring link between $i,j$ robots. The force generated by the linear spring-damper link $sd_{ij} = (sd_{ijx}, sd_{ijy})$ is computed as:

$$sd_{ij} = k_s (d_{ij} - d_{0ij}) d_u + k_v v_{ij}$$

(2)

The constants $k_s$ and $k_v$ are the spring and damping coefficients, $d_{ij}$ and $d_{0ij}$ are the distance between the robots
and the rest distance, respectively, $d_{ij}$ is the unit vector linking $i, j$ robots and $v_{ij}$ is their relative velocity.

The force generated by the torsional spring link $s_{ij} = (s_{ijx}, s_{ijy})$ is computed as:

$$s_{ij} = \frac{k_{ro}(\theta_{ij} - \theta_{0ij})}{d_{ij}} n$$

where $k_{ro}$ is the torsional spring coefficient, $d_{ij}$ is the distance between the robots $i, j$, $n$ is the normal vector on the movement plane and $\theta_{ij}, \theta_{0ij}$ are the angle between robots $i, j$ and the rest angle between them respectively. The constants, $k_s, k_r, k_{ro}$ have been chosen to have a slightly overdamped behaviour.

A force $D_i$ is introduced as a damping term due we want to simulate a real system. It is defined by:

$$D_i = f_d v_i$$

where $f_d$ is the damping coefficient and $v_i = (\dot{x}_i, \dot{y}_i)$ the velocity vector of the robot.

The obstacle avoidance capability is provided by means of the external forces on each robot. This force $E_i$ (see Fig. 2) is generated by the environment and it is applied to the slave robots in the formation. In the next section we explain the process of computing the value of this force, always bounded to a maximum value.

The total force $F_i$ on the $i$ robot of the team is:

$$F_i = G_i + SD_i + D_i + E_i$$

It includes the influence of the spring-damper structure on each robot, the damping force, the force generated by the goal (in the leader robot) and the force generated by the environment that provides obstacle avoidance.

B. Motion Generator

In order to apply to real robots the computed forces $F_i$ to provoke their motion we use a Motion Generator (MG) for differential-drive mobile robots (the ones used in our experiments). The MG transforms these forces into linear and angular velocities according to the equation:

$$\dot{x}_i = Px_i + QF_i$$

where

$$P = \frac{-2b}{mr} \begin{bmatrix} 1 & 0 \\ 0 & k_i d^2 \end{bmatrix}, \quad Q = \frac{1}{m} \begin{bmatrix} 1 & 0 \\ 0 & k_i h \end{bmatrix}$$

By solving this differential equation we can obtain the linear and angular velocities $x_i = (v, \omega)$. This model takes into account the kinematic and dynamic constraints of the robot, so generating feasible trajectories for all the robots in the formation. The parameters $r, d, h$ are geometric constants of the robots (wheel radius, distance between the robot center and the wheel, and moment arm respectively) and $m$ is the mass. More details about the model and how the parameters $b$ (viscous friction) and $k_i$ (inertial coefficient) can be tuned to obtain an overdamped behaviour can be found in [13].

IV. ADAPTATION TO THE ENVIRONMENT

We describe in this section how robots inside a formation use their sensors to perceive the environment, how it is modeled and integrated to reach obstacle avoidance.

A. Sensor data processing

We are using robots equipped with laser rangefinder sensors that compute the distance to the obstacles in a 180 degrees field of view. For the purpose of obstacle avoidance of the robots in the formation, the sensor information has to be filtered and processed. The basic treatments are:

1) All the obstacles that are not close enough, given a security distance to the robot, are discarded.
2) The points inside this security zone are grouped in straight line segments using a split and merge algorithm. The parameters of the segmentation algorithm are tuned depending on the size of the obstacles expected in the environment.
3) An influence zone is computed for every segment. It is defined as the infinite rectangle generated while shifting the segment orthogonally to its own direction. All those segments that keep the robot inside their influence zone are considered (see Fig. 3(a)).

B. Interaction with robots

Each of the obstacles that are influential in robot movement, generates a virtual force which consists in two different components, as shown in fig. 3(b).

- The repulsive component $F_r$ is the one that avoids obstacles. It is defined by equation 8, where $d$ is the orthogonal distance from the robot center to the straight line that holds the segment, $k$ is a tunning parameter used to fuse this force with those concerning structure maintenance and $u$ is a unit vector ortogonal.
to the segment direction; it is always pointing from the segment to outside.

\[ \mathbf{F}_e = \frac{k}{d^2} \mathbf{v} \]  

(8)

- The tangential component \( \mathbf{F}_t \) is used as a guide for the robots to follow the leader. It is parallel to the segment and it is pointing to the direction of the projection of the vector defined from the robot center to the leader of the formation. It helps the robot to follow the leader. The module of the component is defined too by equation 8.

One special case happens when a segment direction is the same than the leader orientation in case of corridors. In this situation, the tangential force is not doing a guiding work but just accelerating the robot towards the leader, so it is not considered.

The total force that the environment induces in each robot \( i (\mathbf{E}_i \) in equation 5), is computed from all the segments that influence each robot.

The \( k \) parameter in equation 8 is used to tune the integration of this environmental force inside formation movement. This tuning process is mostly empirical and the goal is to find such a value that allows obstacle avoidance while formation is maintained and the path is followed.

The \( d \) module of the component is defined too by equation 8.

V. LEADER NAVIGATION AND PLANNING

In section III we have defined how follower robots can keep the formation while moving. In section IV we have defined the influence of the environment in robots movement to permit obstacle avoidance. We present here the control for the leader of the formation. This robot computes the trajectory to the goal and guide the other vehicles to make all the formation reach the objective.

To compute the trajectory, a two-level path planning is considered, one global and other local to the leader robot. The main difference between them is the working scale. While global planning admits low map resolutions, the local needs a highly detailed map to achieve the optimal feasible trajectory. Concretely we are using grid maps with a cell size of, respectively global and local planner, 0.5 m. and 0.05 m.

A. Global Path Planning

The global planner computes the general trajectory of the formation. In other words, given a goal and considering the known environment, it returns a list of waypoints for the formation. Some of the planners with this capabilities need a complete map of the environment to work. But this situation is some kind of unreal, because environments changes dynamically and precomputed maps may become obsolete very quickly. For this reason, we decided to use E* planner [14]. This planner is able to work in unknown environments, can incorporate dynamically new obstacles to the map and can replan the trajectory when changes affect the actual one.

B. Local Path Planning

The local path planner extends obstacle avoidance methods, following the scheme presented in [15]. Usually, these methods are reactive in the sense that they only use the very last information from the environment that a mobile robot has. But some obstacles like U-traps require more information to be avoided. Local path planner gives the robots a memory capacity that permits this kind of manoeuvres.

The requirements for this path planner are different that for the global one. It is needed a moving map, joined to the leader robot, that keeps a local plan. The optimal planning method in this situation is harder to find and deeply depends on the application as commented in [16]. We use D* but A* would fit too.

For obstacle avoidance issues, the leader uses Nearness Diagram (ND) method [15], which gives robot a very accurate trajectory tracking while avoiding static and dynamic obstacles.

VI. EXPERIMENTAL RESULTS

A. Simulations

To test the system, we have made simulations in the multi-robot Player/Stage platform, which reach reasonably similar behaviours to the real platforms.

The first simulation is a trial scenario that result very useful in order to check the system. It represents a narrow corridor with several turns where the initial formation configuration does not fit and some adaptation is required.

In Fig. 4, two different scenes of the simulation are shown. The left one shows how robot enter into the narrow corridor adapting the shape of formation. Figure on the right shows the inverse behaviour. Robots get back the original shape of the formation when they get out of the corridor.

In Fig. 5 followers reactions to leader movements can be seen in linear and angular velocity graphics. The virtual spring-damper structure defined for the formation and the motion generator that computes the velocities make these graphics smooth. Their behaviour is very important in order to ensure that the robots will describe feasible and overdamped trajectories in real experiments.

A complete video of this experiment is attached electronically, so the entire navigation through this scenario can be seen.

In the second experiment the simulation of an evacuation mission, one of the missions foreseen in the URUS project,
is achieved by a team of robots in formation. We use the real
map of the University campus where the real experiments of
the project will be done. Fig. 6 represents four snapshots of
the simulation experiment in different instants. This scenario
begins with the simulation of a fire detection in the square
of the campus. Robots are deployed near the fire to start
guiding people to a given safe place.

We can see in Fig. 6 how leader finds the right trajectory
to the goal and the other robots follow the first one while
obstacles are avoided.

B. Real Experiments

To evaluate our system on a real environment we use
three Pioneer P3-AT from Activmedia. We test the formation
control algorithm through a building corridors (see Fig. 7).
Due to our navigation model needs to exchange information
(localization of all the robots) in order to compute the
movement, we have to use a communication protocol. A
real time wireless multi-hop protocol [12] is used for this
proposal.

The Fig. 8 presents the linear and angular velocities during
the experiment. The velocity graphics of real experiment
show how the leader motion is propagated to the follower
robots. We can see clearly in angular velocity graphic (Fig.
8(b), relevant deviations between cycles 70 and 160. This
is caused by the influence that the narrow zone (see Fig.
7) induces on the formation structure. The linear velocity
reduces when the slave robots cross the narrowing of the
way, while the formation adapts to it.

In some case the velocity profile is not as smooth as in
simulation. The main reason for these deviations are related
to localization, concretely, odometry errors become bigger
when the ground is too polished like in the scenario. In future
experiments the localization errors will be reduced by using
scan matching techniques or SLAM methods developed in
our lab.

We have shown how robot formations can avoid obstacles
adapting its shape to the environment. Moreover, under real
conditions, we have checked that our navigation system fits
the time constraints of the real robot control cycle, which is
strongly related to the time the sensors needs to gather data
(~ 300 ms).

VII. CONCLUSIONS

We have developed a complete navigation system for
robots formations that is able to adapt the shape of the
structure to the environment. This system includes:

- A formation structure definition to keep relative posi-
tions between robots.
- A model of the environment for using it in obstacle
avoidance.
- A two-level path planning and obstacle avoidance in-
tegration to compute trajectory and to improve the
formation leader navigation.
The integration of all these subsystems is flexible enough to cope with the different situations that can be found in real environments. This has been done by tuning the parameters defined in equations to reach an equilibrium in which navigation adapts correctly to the environment, and produces smooth trajectories compatible with the kinodynamic robot constraints.

As future work we will improve the environment model and the cooperative planning to deal with more complex and moving obstacles in more dynamic scenarios.

Another important work is to do the real experiments of the URUS project in real scenarios. The techniques presented in this paper have to be integrated with cooperative perception systems to build a robot network system for applications in urban environments.

REFERENCES