Graph-Based Distributed Control for Adaptive Multi-Robot Patrolling through Local Formation Transformation

Alicja Wasik, José N. Pereira, Rodrigo Ventura, Pedro U. Lima and Alcherio Martinoli

Abstract— **Multi-robot cooperative navigation in real-world environments is essential in many applications, including surveillance and search-and-rescue missions. State-of-the-art methods for cooperative navigation are often tested in ideal laboratory conditions and not ready to be deployed in real-world environments, which are often cluttered with static and dynamic obstacles. In this work, we explore a graph-based framework to achieve control of real robot formations moving in a world cluttered with a variety of obstacles by introducing a new distributed algorithm for reconfiguring the formation shape. We systematically validate the reconfiguration algorithm using three real robots in scenarios of increasing complexity.**

I. INTRODUCTION

Cooperative navigation is a critical feature for Multi-Robot Systems (MRS) undertaking a variety of tasks. The ability to efficiently navigate as a group enables a team of robots to perform activities not possible for single robots. As inclusion of MRS in real-world scenarios and daily life activities increases, so does the need for robust cooperative navigation in increasingly complex indoor environments.

Graph theory offers tools that allow individuals to maintain a desired formation shape in a distributed manner, resulting in a decentralized control of the network of robots, which is more robust than the equivalent centralized architecture because there is no single point of failure [1]. In our work, we employ graph-based distributed control to achieve smooth multi-robot cooperative navigation. We employ a leader-follower strategy to achieve team waypoint navigation. Our underlying single-robot navigation algorithm enables a waypoint navigation of the leader and decentralized, close-vicinity obstacle avoidance for all the robots.

Current state-of-the-art methods allow for sound distributed control and good performance in coordinated navigation. Underlying obstacle avoidance is sufficient to deal with open environments with a sporadic presence of obstacles. However, for higher density of obstacles and cluttered environments, additional complexity is needed. Despite the existence of several methods for obstacle avoidance while navigating cooperatively, most are tested in laboratory conditions or with strong assumptions regarding the placement or sizes of the obstacles. For instance, an attractor dynamic approach with its implicit capacity to split and join formation upon encountering an object is used in [2] to avoid dynamic obstacles. While the attractor dynamic approach limits authority over the formation topology to situations in which no obstacles are present, it nevertheless breaks the formation during obstacle circumvention. Dynamic change of formation has been explored in [3]. Designed for unmanned ground vehicles, this method performs well in open outdoor spaces, but may not be adequately reactive indoors. The decision is taken globally by the leader, which upon detection of an obstacle, sends signals to the followers informing about the new desired formation topology. Formation control using a potential field approach for obstacle avoidance augmented with formation scaling, morphing and rebuilding behaviors has been presented in [4]. After successful obstacle negotiation, a new formation is selected to minimize the total formation error. This approach, however, requires all the robots to participate in the decision making and in the formation topology deliberation; the formation change is global (as opposed to local) and implemented as a switch between two different shapes. Other approaches to formation geometry changes for obstacle avoidance include priority-based arbitration between the behaviors [5], formation change using a transition matrix [6], scaling [7], or avoiding obstacles while treating formation as a rigid body [8]. Deformable formations are studied in [9][10], where exploration of a noisy environmental field drives shape of the multi-sensor formation.

A common assumption in the above approaches is that (a) the experimental area is large, i.e. the formation has enough space to reorganize after negotiating the obstacle, or (b) the obstacles are cluttered in the environment and do not restrict the environment itself, i.e. the experimental settings reflect outdoor, unstructured environments without boundaries. In view of the limitations summarized above, we introduce the Local Formation Transformation (LFT) algorithm, an approach to dynamic formation change that is *local* (meaning that the formation is only reshaped in the immediate neighborhood of the robot that initializes the change) and *gradual* (meaning that the formation does not switch topologies but is modulated in its shape to some extent), with the level of shape alteration proportional to the density of obstacles around the robot. To the authors' knowledge, LFT is the first formation change method implemented on real robots, performed by the robots *locally* and *independently*, in the sense that any robot can alter local formation shape without imposing a global reconfiguration. We relax the

Supported by ISR/LARSyS Strategic Funds from FCT project FCT[UID/EEA/5009/2013] and FCT/11145/12/12/2014/S and FCT/PD/BD/105784/2014 and by European MOnarCH project FP7-ICT-9-2011-601033.

Alicja Wasik, José N. Pereira and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne, Switzerland. firstname.lastname@epfl.ch

Alicja Wasik, Pedro U. Lima and Rodrigo Ventura are with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal. pal@isr.tecnico.ulisboa.pt and rodrigo.ventura@isr.tecnico.ulisboa.pt

traditional assumptions on experimental settings and present results of cooperative formation patrolling in structured indoor areas with characterized by spaces (i.e. spaces not large enough to deploy the desired formation) and several distinct combinations of static and dynamic obstacles.

The long-term goal of this work is to enable cooperative multi-robot navigation in human-populated environments. Our intention is to bridge the gap between human-aware behaviors and multi-robot navigation, where cooperation among the robots and their social interactions with humans are guided by behavioral rules that are acceptable and desirable given the situation. In order to be useful in human-populated environments, multi-robot navigation needs to: 1) navigate smoothly around obstacles, 2) be flexible in terms of change as to encompass the aforementioned behavioral rules, and 3) be visually easy to read so that the intentions of the formation can be properly interpreted by humans. Consequently, many of our design choices were dictated for that purpose, including heading alignment, fluent formation change, and the choice of large, holonomic robots as validation platform.

In this paper, we present experiments with three real robots performing a cooperative patrolling task in multiple scenarios designed to represent cluttered environments of incremental difficulty. We have defined dedicated metrics for performance evaluation throughout the trials, while a verifiable measure of success is the ability of the formation to smoothly navigate in indoor areas without collisions or need of human intervention.

This paper is organized as follows. In Section II we provide an overview of the control methodology. In Section III we describe the Local Formation Transformation algorithm. Experiments with real robots are presented in Section IV. We draw conclusions in Section V.

II. CONTROL METHODOLOGY

A. Graph Theory and Laplacian Control

An undirected graph with N elements is defined as a pair $G = (V, E)$, where $V = \{v_i, i = 1 \dots N\}$ is the *vertex set* and $E \subseteq V \times V$ is the *edge set*. Edges represent the flow of information, that is if v_i can observe v_j, then there exists an edge $e_{i,j}$. In this work if v_i can observe v_j then v_j can also observe v_i, thus the edges are undirected. In order to form a valid *incidence matrix* $\mathcal{I} \in \mathbb{Z}^{N \times |E|}$ which is defined on directed graphs, we assign a random orientation to the edges of the undirected graph G. The incidence matrix describes which edges connect which nodes and takes the following values:

$$\mathcal{I}_{ik} = \begin{cases} -1 & \text{if } e_k = (n_i, n_j) \\ 1 & \text{if } e_k = (n_j, n_i) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $|E|$ is the cardinality of the edge set and e_k is the k^{th} edge of G. In this work, vertices correspond to the robots. Edges are labeled by the Euclidean distances between the robots. The graph is considered fully connected because each robot maintains the information about the spatial states of the others at all times.

Consider the robots modeled using a single kinematic integrator, $\dot{p}_i = u_i$, where $p_i \in R^2$ represents i^{th} robot's position and u_i is a control input. For a team of N robots, a solution to the rendezvous problem (all robots converging to the same position) can be solved with the *Laplacian feedback* control law $\dot{p} = -\mathcal{L}p$ [1]. *The Laplacian matrix* $\mathcal{L} \in R^{\text{N x N}}$ defined as $\mathcal{L} = \ \mathcal{I}\ \mathcal{I}^T$ for an undirected graph is symmetric and positive semi-definite [11], with one eigenvalue equal to zero. The zero eigenvalue assures state convergence $lim_{t \to \infty} p(t) = v_1 v_1^T p_0$, where $v_1 = \frac{1}{\sqrt{N}}$ is the normalized eigenvector corresponding to the zero eigenvalue. While $\dot{p} = -\mathcal{L}p$ drives the agents to a rendezvous, formation control requires an additional bias matrix b, so that:

$$\dot{p} = -\mathcal{L}(p - b) \tag{2}$$

defines a desired deviation from a center point of the formation [1]. A *weighted Laplacian* is a Laplacian with weights assigned to the edges: $\mathcal{L} = \ \mathcal{I}\ \mathcal{W}\ \mathcal{I}^T$, where $\mathcal{W} = diag(\{w_k, \forall e_k \in \ E\}) \in \mathbb{R}^{|E| \times |E|}$ is the weight matrix.

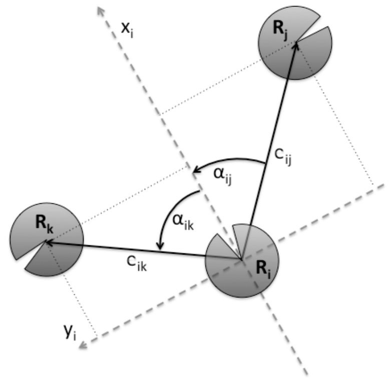

Fig. 1: Illustration of the graph-based connectivity information with respect to the robot R_i. c_{ij} is the Euclidean distance between R_i and its neighbor R_j, and α_{ij} is the bearing. Dashed lines indicate the local coordinate frame of R_i.

B. Active Formation Based on Leader Networks

Leader networks extend the idea behind the graph-based control by allowing some group members to perform an independent behavior, while the others execute the consensus algorithm underlying the Laplacian feedback control law. Specifically, the graph $G = (V, E)$ becomes $\tilde{G} = (\tilde{V}, \tilde{E})$, where the vertex set $\tilde{V}$ holds the *leaders subset* V_l and the *followers subset* V_f. Partition of the vertex set leads to an analogous subdivision of the edges, $\tilde{E} = E_l \cup E_f \cup E_{fl}$. The weight matrix is partitioned so that changing weights on some particular edges triggers a desired group dynamics. Details of how to partition the Laplacian matrix to encompass dissimilarities between leader and follower vertices can be found in [12].

The relative strength of the connection edge, given by the magnitude of the associated weight, determines the convergence rate towards the bias. In this work, weights have been tuned according to the distance c_{iL} between the follower R_i and the leader L using functions $w_{ff,\, i}(c),\ \forall e_i \in E_f$ and $w_{fl,j}(c), \forall e_j \in E_{fl}$, as proposed in [13]. If c_{iL} is large, the weight of the follower-follower edge w_{ff} is smaller than on the leader-follower edge w_{fl} and the followers have a higher potential to reach the leader first. If the opposite is true, the followers are forced to converge to the formation first.

C. Graph-based Formation Control

In this work, we assume that the robots are holonomic and that each of them has access to the state information of each robot R_i, comprised of the pose $\bar{\boldsymbol{p}}_i = [x_i, y_i, \phi_i]^T$.

For a robot R_i, the formation control presented in Equation (2) is achieved as follows:

$$\begin{aligned} \bar{c}_{x,i}(t) &= \tfrac{1}{\Delta_i+1} \textstyle\sum_{j=1}^{\Delta_i} \left[-\mathcal{L}_{ij}(c_{ij}(t)\cos(\alpha_{ij}(t)) - b^x_{ij}(t))\right] \\ \bar{c}_{y,i}(t) &= \tfrac{1}{\Delta_i+1} \textstyle\sum_{j=1}^{\Delta_i} \left[-\mathcal{L}_{ij}(c_{ij}(t)\sin(\alpha_{ij}(t)) - b^y_{ij}(t))\right] \end{aligned} \tag{3}$$

where Δ_i is the number of neighbors of robot R_i, i.e the robots connected by an edge to R_i, c_{ij} and α_{ij} are the Euclidean relative position vector and the bearing between the robots R_i and R_j respectively (see Figure 1). The bias matrices b^x_i, $b^y_i \in \mathbb{R}^{\Delta_i}$ define the desired inter-robot distances in the leader's reference frame. The result is a proportional controller for the i^{th} holonomic follower:

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = K_u \begin{bmatrix} \bar{c}_{x,i}(t) \\ \bar{c}_{y,i}(t) \end{bmatrix} \tag{4}$$

where K_u is a positive constant. Comprehensive demonstration of this controller stability has been presented in [14].

Finally, we decouple the follower's heading control, whose objective is to match the orientation of the leader:

$$\dot{\phi}_i = v_\theta = K_\phi(\phi_L - \phi_i) \tag{5}$$

where K_ϕ is a positive constant and ϕ_L is the orientation of the leader. Matching the follower orientation with that of the leader has been chosen for being visually more readable by humans.

D. Navigation Methods

In our work, the strategy employed for cooperative navigation comprises decoupling formation control from reactive obstacle avoidance. While the former is handled at team level, using a graph-based approach as described above, the latter is handled at the individual level and has priority. That is, each robot performs collision-free trajectories using its range sensors. Note that, even though formation control does not take into account obstacles, their influence in the individual robot trajectories is fed back into the formation controller. For instance, if a robot is forced into a different trajectory to avoid an obstacle, that change is propagated through the graph in real time to the other robots.

The obstacle avoidance method builds upon previous work on waypoint navigation [16]: given a goal location, a Fast Marching Method (FMM) is used to obtain a local-minima-free potential field encoding the optimal direction of motion towards the goal, for any given point (provided that the goal is reachable from that point) followed by a Dynamic Window Approach (DWA) method to perform obstacle avoidance while driving the robot along that optimal direction. This is done by casting the problem of determining the next actuation command into the following (schematic) optimization problem:

$$\underset{\bar{v} \in \mathcal{A}}{\text{maximize:}} \quad a\,P(\bar{v}) + b\,C(\bar{v}) \tag{6}$$

where P and C are functions of velocity candidates $\bar{v} = (v_x, v_y, v_\theta)$ (in the robot frame) that evaluate progress P towards goal (according to the potential field obtained from the FMM) and clearance C from obstacles, respectively, with $\mathcal{A} \subset \mathbb{R}^3$ being the set of valid actuation candidates (see [16] for further details). Weights a and b specify the relative importance of each one of these evaluation functions to the combined cost function. The set of valid candidates $\mathcal{A}$ is obtained from an equally spaced grid of points in actuation space $\hat{v}$, constrained to (1) speed and acceleration limits of the robot and to (2) the guarantee that, for each candidate $\bar{v} \in \mathcal{A}$, the robot can stop at maximum deceleration with the same velocity direction before hitting a perceived obstacle. Note that perceived obstacles include the other robots in the formation, as well as any other obstacle present in the environment. The formation control method delivers, for each follower robot i, a velocity reference u_i. The obstacle avoidance method described above integrates this information and is modified in the following way:

$$P(v) = -\frac{||\bar{v} - u_i||}{V_{MAX}} \tag{7}$$

where V_{MAX} is the specified maximum robot speed, used here to normalize this function. Thus, the DWA will tend to prefer velocity candidates closer to the velocity reference u_i, while assuring collision-free trajectories because of the candidates $\bar{v}$ being obtained from the set $\mathcal{A}$. $\mathcal{A}$ is a hard constraint to ensure collision-free trajectories, while the C term in Equation (6) promotes soft clearance to obstacles (e.g., a narrow door).

III. LOCAL FORMATION TRANSFORMATION

One can envision that when navigating in a free open space, whether for coverage or sensing purposes, a team of robots should be spread out, while in cluttered environments a tight formation may be preferable. It may be beneficial to let the team autonomously modify the formation in reaction to environmental changes and different environmental conditions, in terms of available area, topology of that area and density of obstacles. We propose a solution to the problem of navigating in structured, confined and cluttered spaces by extending the graph-based formation control method with the capability to dynamically alter the formation shape depending on the characteristics of the environment. In this section we first describe the principles behind the LFT algorithm, followed by a formal definition and details of the implementation.

A. Approach

Intuitively, the LFT algorithm works in the following manner. We define a variable η_i that reflects a local density of obstacles with respect to robot R_i and drives its gradual (and local) formation change. Two extreme circumstances can be distinguished: A) when there are no obstacles around the follower, $\eta_i = 0$ and the robot should remain in the defined place in the formation; B) in case of maximal density of obstacles, $\eta_i = 1$, the safest path for the follower is to be aligned within a parametrized tolerance margin behind the leader. Under such circumstances we can assume its path to be collision free. In particular, this is the case for sufficiently

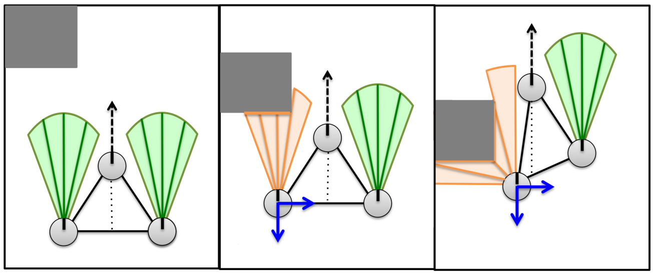

Fig. 2: Illustration of the Local Formation Transformation algorithm. (Left) The default formation. (Middle) The follower on the left detects an obstacle and starts gradual alignment behind the leader. (Right) The follower on the left remains partially aligned behind the leader until the obstacle is cleared.

static environments when the leader's heading is constrained to be tangential to its trajectory by a motion planning module.

The variable η_i varies continuously between 0 and 1, and thus the follower gradually wavers between its desired place in formation and the safe place behind the leader. The estimated obstacle density $\eta_i \in [0, 1]$ serves as a sole indicator of how the follower should modify the original formation matrix to navigate around the obstacles. It is computed based on a constrained virtual sensor field of view characterized by a sensing range γ_i and a sensing angle σ_i. The field of view (γ_i, σ_i) covers an area to be traversed in the near future, with an exception when $\eta_i \rightarrow 1$ and the robot queries the area around the desired place in the formation to return there when it is safe to do so.

B. *Definition*

Because of the continuous adaptation of the follower R_i to maintain its desired relative position with respect to the leader, the resulting formation change can be formalized as a transformation function in a D-dimensional space $\psi_i : [0, 1] \rightarrow \mathbb{R}^{\Delta_i \times D}$, $\forall i \in V_f$:

$$S = \langle \psi_i(0), \psi_i(1), \psi_i : \eta_i \rightarrow \mathbb{R}^{\Delta_i \times D} \rangle \tag{8}$$

That is, given two geometries $\psi_i(0)$ and $\psi_i(1)$, we design a smooth, continuous function of class C^1 that maps the density of obstacles perceived by the robot R_i in the environment to the modification of the formation geometry $\psi_i \in [0, 1]$ (an example is shown in Figure 3). The geometry $\psi_i(0)$ specifies the original shape of the formation and is assumed to be determined by a higher level controller or external user. The purpose of $\psi_i(1)$ is to define a virtual column formation shape and specify an allocation of the robots within such structure. Using $\psi_i(1)$ guarantees that each follower has a dedicated place behind the leader in the extreme case when $\forall i \in V_f$, $\eta_i \approx 1$ and all followers must fall behind the leader.

Remark 1: The specification of the LFT algorithm requires the transformation function ψ_i to transform the shape of the formation from $\psi_i(0)$ to $\psi_i(1)$. Consider the specification of the formation in Equation (8). If the transformation function is designed with the constraint:

$$\forall i, j \in V_f, \parallel \psi_i(\eta_i) - \psi_j(\eta_j) \parallel \geq \epsilon \tag{9}$$

where ϵ is the minimum allowed distance between any two robots (e.g., the robot diameter plus a safety margin), then collisions between the members of the formation will not occur. The algorithm is generalizable to N robots and different types of formation, because it is possible to design $\psi_i(1)$ and $\psi_i : [0, 1] \rightarrow \mathbb{R}^{\Delta_i \times D}$ so that the transformation between two formation shapes is collision-free. An illustrative example of the transformation function ψ_i is shown in Figure 3.

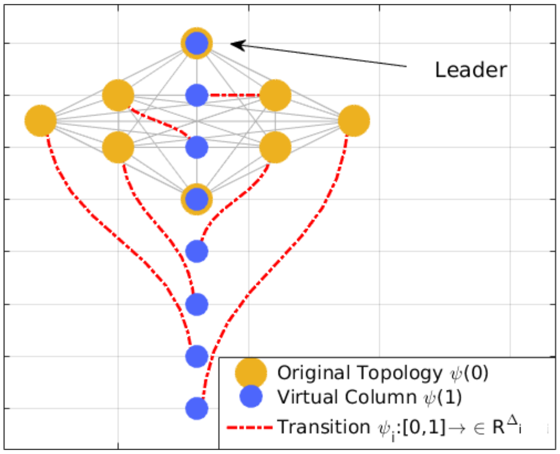
Leader
Original Topology $\psi(0)$
Virtual Column $\psi(1)$
Transition ψ_i:[0,1]→ ∈ R$^{\Delta_i}$

Fig. 3: Illustration of the transition function ψ. The desired formation of the robots is $\psi_i(0)$ (orange), upon detection of obstacles they transition to a column topology $\psi_i(1)$ (blue).

C. *Implementation*

While the abstract definition of LFT in Section III-B does not restrict the algorithm to a specific formalization of formation control, here we present its realization within the graph-based framework. Particularly, the transition function $\psi_i(\eta) : \eta_i \rightarrow \mathbb{R}^{\Delta_i \times 2}$ corresponds to the bias matrices b_i^x, $b_i^y \in \mathbb{R}^{\Delta_i}$ defined in Section II-C as follows:

$$b_{iL}^{x,y\prime} = \psi_i(\eta_i) : [0, 1] \rightarrow \mathbb{R}^{\Delta_i \times 2} \tag{10}$$

where L denotes the leader. The bias to the leader is changed on two axes simultaneously: by modifying b_{iL}^x the follower R_i aligns behind the leader, while by modifying b_{iL}^y it follows at a closer or further distance from the leader. The functions for changing $b_{iL}^{x,y}$ were chosen as follows:

$$\begin{aligned} \psi_i^x(\eta_i) &= b_{iL}^x(1 - 1/(1 + e^{-k_1(\eta_i - k_2)})) \\ \psi_i^y(\eta_i) &= sign(b_{iL}^y)[(\psi_i(0) - abs(b_{iL}^y))\eta_i + b_{iL}^y \end{aligned} \tag{11}$$

where k_1 and k_2 are constant parameters normalizing and attenuating the shape of a function that filters η_i. Thus, the value of $\psi_i(\eta_i)$ updates the bias, and in our implementation takes place every time step $T = 100\ ms$.

Remark 2: The convergence rate of a system based on the Laplacian consensus feedback can be calculated depending on the eigenvalues of the Laplacian matrix $\mathcal{L}$ [11]:

$$\parallel x(t) - \mathbf{1} \parallel \leq \parallel x(0) - \mathbf{1} \parallel e^{-\lambda_2 t} \tag{12}$$

where $\mathbf{1} = [1, ..., 1]^T$ and λ_2 is the lowest non-zero eigenvalue of $\mathcal{L}$ that determines the speed of convergence. Under the assumption of the time-scale separation principle [15] and the constraint that the function ψ is chosen so that $\dot{\psi} \leq \tau e^{-\lambda_2}$, where $\tau \geq 1$, the system described in Equations (4) and (5) is assumed to achieve consensus.

Fig. 4: Robots used in the experiments.

The functions in Equation (11) satisfy the constraints discussed in *Remark 1* and *Remark 2*. They have been tested in simulation with up to 10 robots, but their generalization to a larger number of robots is still to be proved.

The updated bias ψ_i of the follower R_i is shared with all its neighbors. Knowing $\psi_j(0) = b_{jL}^{x,y}$ of neighbor R_j and the new value, $\psi_j(\eta) = b_{jL}^{x,y\prime}$, we can update the bias between R_i and R_j. This step could be omitted in sparse, minimum degree graphs were each follower is only connected to its local leader (for an example of such a formation refer to [17]). A solution not requiring communication and yet retaining a full connection graph could project a virtual robot $\tilde{R_j}$ at a default bias, so that $b_{ij}^{x,y\prime} \leftarrow b_{ij}^{x,y} + (b_{iL}^{x,y\prime} - b_{iL}^{x,y})$, $\forall\, R_j \in V_f, i \neq j$.

LFT complements obstacle avoidance in the sense that it reacts at farther distances through its multi-robot perception capabilities and has an implicit predictive component that estimates a collision course by monitoring the area to be visited shortly in time. While obstacle avoidance prevents collisions in a reactive manner, LFT allows navigation around structured obstacles such as corners or through narrow passages. It implicitly promotes a line formation, as superior for the teams of robots going through tightly confined areas (as proved in [4]).

IV. EXPERIMENTS

Experiments have been performed using MBot robots (Fig. 4) developed within the FP7 European project MOnarCH (Multi-Robot Cognitive Systems Operating in Hospitals) with the goal of introducing social robots in the pedriatric wing of a hospital[1]. The robot is equipped with navigation, perception and low-level safety sensors. For navigation, localization and obstacle avoidance, it fuses measures provided by two laser range finders Hokuyo URG-04LX-UG01 providing 360° field-of-view, odometry encoders and IMU sensors. Mecanum wheels provide an omnidirectional locomotion system with a max speed of 2.5 m/s. The robot radius is 0.6 m. A complete description of the MBot robot can be found in [18]. All algorithms and methods are supported by the ROS middleware.

A. Communication

Robots connect to a local wireless network and exchange messages using an information sharing framework - designated as the Situational Awareness Module (SAM) - that provides the robot with a consistent global picture of the whole system [19]. Each robot running an instance of SAM on its ROS Master can retrieve information from other network nodes and share its data by writing to the SAM repository, where the shared contents is automatically synchronized via multicast routing provided by a third-party *multimaster_fkie package*[2]. In this manner, each robot can communicate the data to other team members.

B. Performance Metrics

The performance of our system is analyzed based on three separate criteria to be minimized:

$$
\begin{aligned}
E_D &= \frac{1}{\|b_D\| N_F} \sum_{i=1}^{N_F} \left(\frac{1}{|\, \mathcal{N}_i \,|} \sum_{R_j \in \mathcal{N}_i} \Big| \|\bar{\boldsymbol{p}}_j - \bar{\boldsymbol{p}}_i\| - \|b_{ij}^{xy\prime}\| \Big| \right) \\
E_O &= \frac{1}{\pi N_F} \sum_{i=1}^{N_F} \Big(|\phi_i - \phi_{L(i)}| \Big) \qquad (13) \\
E_A &= \frac{1}{N_F} \sum_{i=1}^{N_F} (\eta_i)
\end{aligned}
$$

averaged over the time of the experiment, where N_F is the number of followers and $\mathcal{N}_i$ is a set of neighbors of robot R_i. E_D assesses the formation shape by verifying how far inter-robot distances are from those specified with the bias matrix. E_O evaluates the orientation control by verifying differences in heading alignment between leaders and followers. E_A has been chosen as an indirect measure of the LFT reactivity to avoiding obstacles. The rationale behind this choice is that as soon as η starts to increase, the robot modifies locally its bias to lessen the chance of driving close to obstacles, leading in turn to a reduction of η. The selected performance metrics enforce the desirable properties of the formation that were motivated by operation in human-populated environments: E_D and E_O evaluate the quality of the formation and its visual readability, while E_D and E_A the navigation smoothness around the obstacles.

C. Experimental Setup

Experiments are performed in two indoor environments shown in Figure 5 and Figure 14. Particularly, our patrolling scenarios take place in two rooms connected by a corridor. The setting is cluttered with various appliances, including furniture, lab equipment, and structural building features such as doors and columns. The objective is to assess impact of environmental peculiarities that have the highest effect on the formation. The robots positions are obtained from the robot self-localization using the *AMCL*[3] package from ROS.

[1] MOnarCH, FP7, FP7-ICT-2011-9-601033 (http://monarch-fp7.eu)

[2] multimaster_fkie (http://wiki.ros.org/multimaster_fkie)

[3] AMCL (http://wiki.ros.org/amcl)

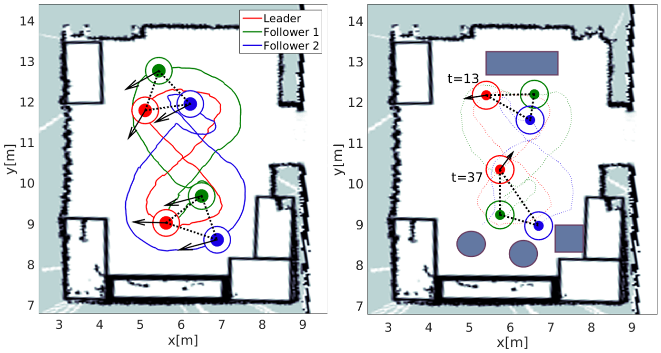

Fig. 5: (Left) Trajectories of the robots in the scenario S_I. (Right) Trajectories of the robots in the scenario $S_{II}(A)$ with static obstacles. Squares located around the arena are the desks and other furniture. Size of the outer circles indicates the bounding box of the robot.

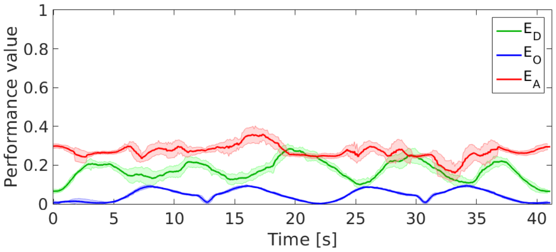

Fig. 6: Performance evolution for the scenario S_I. Light colors indicate standard deviation averaged over 10 runs and the two followers. This convention remains valid for the performance evolution plots throughout the paper.

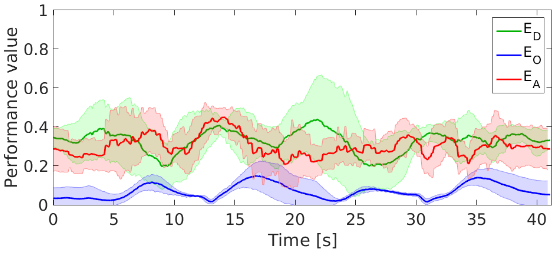

Fig. 7: Performance evolution for the scenario $S_{II}(A)$.

D. Scenarios

We investigate the performance of the LFT algorithm under various working conditions in four different scenarios and six experiments. For each experiment we perform 10 runs with initial positions of the robots at the same locations. We consider a triangular formation with two followers and one leader forming an equilateral triangle with 1 m sides. The parameters used in the experiments are: $a = 1$, $b = 10$ and $K_u = K_\phi = 0.8$.

1) S_I – Trajectory Twist Scenario: This experiment tests the ability of the formation to move on a convoluted trajectory with sharp turns and serves as a baseline for performance comparison. The leader moves at a speed $\sim 0.5 m/s$ in an eight-figure trajectory (see Figure 5, (left)), which is retained for scenarios S_I to S_{III}.

2) S_{II} – Static Obstacles Scenarios: In this scenario we test the LFT algorithm in three settings with the arena cluttered with static obstacles positioned so as to vary the difficulty of the experiment and illustrate when the algorithm anticipate the reactive obstacle avoidance.

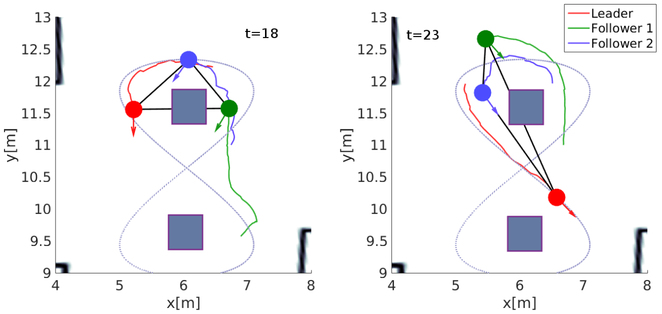

Fig. 8: Snapshots of the experiment $S_{II}(B)$.

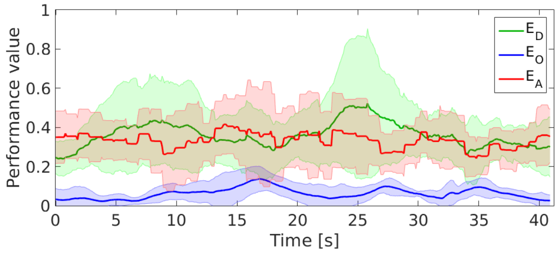

Fig. 9: Performance evolution for the scenario $S_{II}(B)$.

$S_{II}(A)$ – In the first setting the obstacles are scattered at the outer edges of the leader's trajectory (Figure 5 (right)), at positions that require the followers to alter their behavior as they cope with sharp turns.

$S_{II}(B)$ – The second setting (Figure 8) requires the followers to deviate from their original formation as they pass over the inner circle of the eight-figure trajectory.

$S_{II}(C)$ – The third setting evaluates the ability of the formation to pass through a narrow passage of 1.5 times the diameter of the robot (Figure 10).

3) S_{III} – Dynamic Obstacles Scenario: The third scenario S_{III} is designed as to reveal behavior of the formation encountering a dynamic obstacle. The experiment is conducted as follows: we ask a human volunteer to act as a moving obstacle by systematically traversing the experimental arena, and for repeatability of the experiment, follow a path that has been delineated on the floor and carefully timed with a stopwatch. The waypoints of the human are chosen as to impose an action on the follower, when the human crosses its path.

4) S_{IV} – Complex Environment Scenario: The final demonstration of the algorithm in a realistic indoor environment. We consider a waypoint patrolling task designed so that it is necessary for the robots to change the formation during the run as well as rotate in a highly constrained space. The storyline is the following: the leader is to go out of the 1^{st} and the 2^{nd} *door*, pass through a *narrow passage* and then return to the initial position (see Fig. 14). The final waypoint given to the leader is the same as the initial waypoint.

E. Results

S_I – The snapshots presented in Figure 5 (left) show that the robots converge to the desired formation and maintain it. The performance metrics presented in Figure 6 lead to

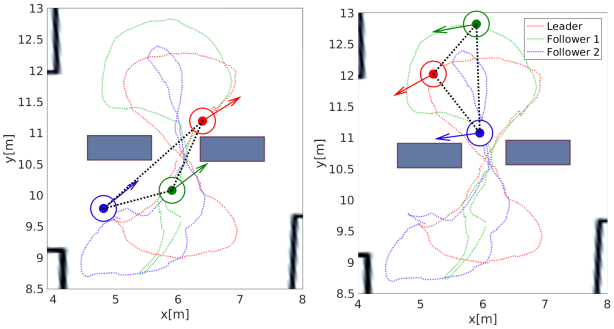

Fig. 10: Snapshots of the experiment $S_{II}(C)$.

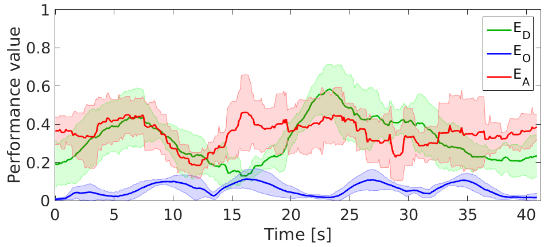

Fig. 11: Performance evolution for the scenario $S_{II}(C)$.

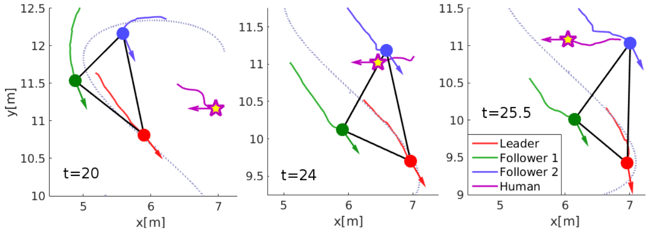

Fig. 12: Snapshots of the experiment S_{III} with the formation being interrupted by a person walking around the arena.

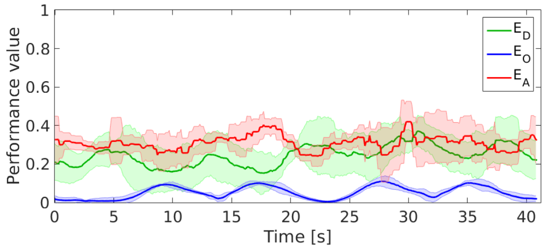

Fig. 13: Performance evolution for the scenario S_{III}.

the conclusion that the intricacy of the leader's trajectory has no significant impact of on the formation errors. Because of the size of the MBot robot, small rises of E_D during sharp turns are not visible in reality ($E_D = 0.27$ is the average formation error, less than half the robot radius). A small variation of performance over the runs suggests a strong repeatability of the results.

$S_{II}(A)$ – The trajectories shown in Figure 5 (right) reveal how the followers during negotiation of the obstacles locally modify the formation using LFT. The performance presented in Figure 7 only marginally decreases compared to S_I, but has a larger variance because small variations in robots' location or perception reflects in diverse behavior of the LFT.

$S_{II}(B)$ – We observe that the robots cope with the situation in Figure 8 by going around the obstacle on either of its sides, thus two different modalities are revealed. Note that this situation is more cumbersome than $S_{II}(A)$ because of the obstacle combined with a sharp turn. This is reflected by a decrease of performance, especially of the E_D component, which attains a peaky performance drops, as illustrated by in Figure 8 (right). The existence of two modalities drives the large variance of performance.

$S_{II}(C)$ – The followers detect obstacles on their path, change to a column formation using the LFT algorithm and, by the means of the DWA obstacle avoidance, avoid any potential collisions that could result from a sharp turn (see Figure 10). The two peaks of E_D in Figure 11 correspond to the two times the robots navigate through the narrow passage, but even then the mean E_D error is smaller than the robot diameter.

S_{III} – The snapshots presented in Figure 12 show the situation before, during, and after the formation encounters a dynamic obstacle. Notice that collisions with swiftly moving obstacles are primarily handled by the virtue of DWA, being more reactive that LFT. Even though the DWA does not guarantee finding an optimal velocity, it prevents collisions by stopping the robot before hitting the obstacle. The performance is little affected as compared to S_I because of the instant nature of obstacle negotiation (see Figure 13).

S_{IV} – Figure 14 shows the trajectories of the robots. The team starts in a triangular formation, which is then modified by the means of the LFT, particularly discernible as the formation passes through the narrow passage at $t = 48\ s$. At the furthest waypoint ($t = 56\ s$), the formation has to rotate around the leader in a confined area with the diameter of the free space smaller than the diameter of the desired formation. As seen in Figure 15, the formation error indicated by the E_D is large at times $t = [38\ s, 56\ s, 81\ s, 98\ s]$, which correspond to the robots passing the 2^{nd} *door*, *narrow passage*, the 2^{nd} and the 1^{st} *door* (see Figure 14). The larger standard deviation corresponds to the time variation over the runs generated by the maneuvers of the followers while going through the narrow areas. Note that without the LFT, the followers, remaining at original positions within the formation, would not have been able to negotiate a passage that has an abrupt entry (such as door leading out of a room) and a diameter smaller than the diameter of the formation. They would head towards the barrier and for wall-type obstacles, they would first stop and then oscillate in parallel to the obstruction, so that as the leader moves away, the formation would break. For smaller, convex obstacles, DWA allows the followers to safely navigate around the obstacles.

The results indicate a high correlation between the density of the obstacles (reflected by E_A) and the formation shape error (E_D). The reason is that if an obstacle momentarily hinders the follower's movement, its connection edges to other robots are automatically stretched. The orientation error remains largely unaffected by the experimental settings, mostly due to the fact that the heading control of the followers is decoupled from the position control.

V. CONCLUSIONS

In this work, we presented an adaptation of the graph-based control framework to enable a dynamic change of formation.

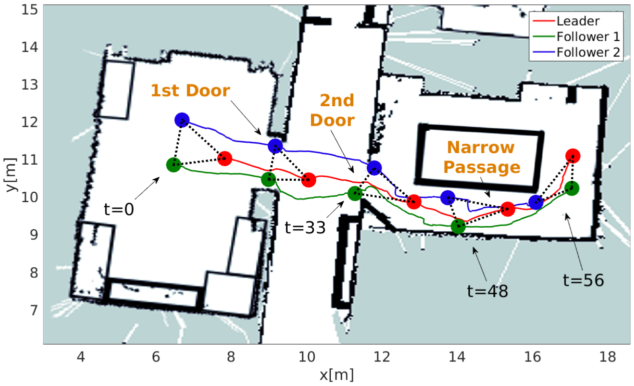

Fig. 14: Trajectories of the robots during the scenario S_{IV}.

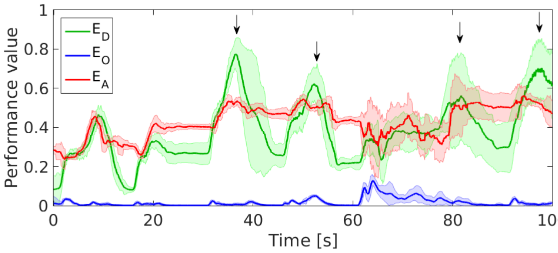

Fig. 15: Performance evolution for the scenario S_{IV}.

We presented a novel algorithm, the Local Formation Transformation, which, depending on the density of the obstacles sensed in the environment, allows the followers to modify the bias matrix, an thus the formation shape locally. We concluded that the LFT enables the formation to navigate as a unit through demanding environments, such as narrow passages with abrupt entry points and the width smaller than the diameter of the formation. Fast formation reconfiguration resulted in the ability of the team to cope with complex building features such as doors or other confined spaces. Moreover, the robots navigated naturally in an environment with human beings, avoiding them smoothly by virtue of obstacle avoidance. Not only were the robots able to perform a long duration cooperative navigation in formation, but they were doing so in an environment with numerous uncertainties arising from the presence of static and dynamic obstacles as well as sensor and actuator errors. Motivated to be used in human-populated environments, the LFT algorithm achieved desirable properties, including smoothness of motion and aesthetic negotiation of obstacles.

We have presented four scenarios investigating the process of maintaining a formation and modifying its shape in environments with various complexities. The experiments served for evaluation of the formation robustness to sophisticated trajectories of the leader, static and dynamic obstacles, and complex indoor environments. Our experimental results compare performance of the formation in the different scenarios, which we supported with a discussion of potential limitations of the algorithm.

Our future research will address i) further improvements of the Local Formation Transformation algorithm, especially a relaxation of its current dependence on communication by introducing virtual follower neighbors; ii) a dynamic formation role assignment, which by optimizing selection of a temporary leader would allow the formation to escape local minima that are not taken into consideration in our approach; iii) an exploration of graph properties allowing the team to explicitly split and regroup the formation, further increasing the agility and the autonomy of group navigation; iv) a formal analysis of the algorithm, focusing on stability and convergence properties.

VI. Additional Material

Video accompanying this paper can be found at: `http://disal.epfl.ch/research/InstitutionalRoboticsFormations`

References

[1] Mesbahi M. and Egerstedt M., *Graph theoretic methods in multiagent networks*, Princeton University Press, 2010.

[2] Monteiro S. and Bicho E., ”Attractor dynamics approach to formation control: theory and application”, *Autonomous Robots*, 29: 331-355, 2010.

[3] Vilca J., Adouane L. and Mezouar Y., ”Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration”, *Proc. Int. Symp. on Distributed Autonomous Robotics Systems (2014)*, Springer Tracts in Advanced Robotics, vol. 112, pp. 237-254, 2016.

[4] Seng W. L., Carlo B. J. and Sekercioglu Y. A., ”Distributed formation control of networked mobile robots in environments with obstacles”, *Robotica*, 34(6):1403-1415, 2016.

[5] Kuppan, C. R., Singaperumal, M., and Nagarajan, T., ”Distributed planning and control of multirobot formations with navigation and obstacle avoidance”, *Recent Advances in Intelligent Computational Systems*, pp. 621-626, 2011.

[6] Desai J. P., Ostrowski J. P., and Kumar V., ”Modeling and control of formations of nonholonomic mobile robots”, *IEEE Trans. Robotics and Automation*, 17(6):905-908, 2001.

[7] Shames, I., Deghat, M., and Anderson, B. D, ”Safe formation control with obstacle avoidance”, *Proc. of the 18th IFAC World Congress*, IFAC Proceedings Volumes, vol. 44, pp. 11252-11257, 2001.

[8] Brandao, A. S., Sarcinelli-Filho, M., Carelli, R., and Bastos-Filho, T. F., ”Decentralized control of leader-follower formations of mobile robots with obstacle avoidance”, *Proc. of the IEEE Int. Conf. on Mechatronics*, 2009, DOI 10.1109/ICMECH.2009.4957167 (6 pages).

[9] Zhang, F. and Leonard, N. E., ”Cooperative filters and control for cooperative exploration”, *IEEE Trans. Automatic Control*, 55(3):650-663, 2010.

[10] Soares J. M., Aguiar A. P., Pascoal A. M., and Martinoli A., ”A distributed formation-based odor source localization algorithm: design, implementation, and wind tunnel evaluation”, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1830-1836, 2015.

[11] Royle G. and Godsil C., *Algebraic graph theory*, Springer Graduate Texts in Mathematics, vol. 207, 2001.

[12] Egerstedt M., Ji M. and Muhammad A., ”Leader-based multi-agent coordination: controllability and optimal control”, *Proc. of the American Control Conf.*, pp. 1358-1363, 2006.

[13] Falconi R., Gowal S. and Martinoli A., ”Graph-based distributed control of non-holonomic vehicles endowed with local positioning information engaged in escorting missions”, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3207-3214, 2010.

[14] Falconi R., ”Coordinated Control of Robotic Swarms in Unknown Environments”, Ph.D. dissertation, Università di Bologna, 2009.

[15] Mease, K. D., S. Bharadwaj, and S. Iravanchy, ”Timescale Analysis for Nonlinear Dynamical Systems”, *Journal of Control, Guidance and Dynamics*, 26(2): 318-330, 2003.

[16] R. Ventura and A. Ahmad, ”Towards optimal robot navigation in urban homes”, *Proc. of the 18th RoboCup Int. Symposium*, 318-331, 2014.

[17] Fredslund J. and Mataric M. J., ”A general, local algorithm for robot formations”, *IEEE Trans. Robotics and Automation*, 18(5):837-846, 2002.

[18] Messias J., Ventura R., Lima P., Sequeira J., Alvito P., Marques C. and Carrico P., ”A robotic platform for edutainment activities in a pediatric hospital”, *Proc. of the IEEE Int. Conf. on Autonomous Robot Systems and Competitions*, pp. 193-198, 2014.

[19] Messias J., Ventura R., Lima P., Sequeira J., *ROS in the MOnarCH Project: A Case Study in Networked Robot Systems*, Springer Book on Robot Operating System (ROS) - The Complete Reference, vol. 1, pp. 375-395, 2016.