Graph Based Distributed Control of Non-Holonomic Vehicles Endowed with Local Positioning Information Engaged in Escorting Missions

Riccardo Falconi, Sven Gowal, Alcherio Martinoli

Abstract— Using graph theory, this paper investigates how a group of robots, endowed with local positioning (range and bearing from other robots), can be engaged in a leader-following mission whilst keeping a predefined configuration. The possibility to locally change the behaviors of the follower team to accomodate both tasks is explored. In particular, a methodology to automatically adjust the parameters of the inter-robot interactions and a nonlinear PI controller are explained and implemented. Our approach is supported by a mathematical analysis as well as real robot experiments.

I. INTRODUCTION

The problem of driving a multi-agent system to a final common state is known as the *consensus problem* and is based on the idea of using some information from the communication network to drive the system to a final state. Indeed, as explained in [1], formation control can be achieved with graph-based theory on holonomic agents.

When solving the rendez-vous problem with non-holonomic robots, most works, [2], [3], assume the availability of global positioning (such as a compass) and general knowledge about all the other robots and the environment. Unfortunately in real-life scenarios, this general knowledge can rarely be acquired or computed at high enough rates to enable consistent and correct formation keeping. Furthermore, leader-following strategies should not rely on these assumptions since the leader should be independent from its followers. In this paper, the only information available to each agent, as explained in Section II-C, is the relative range and bearing of the other robots when available.

Using limited perception or localization abilities to create robotic formations has received recent attention in [4], [5], [6], [7], [8]. In our previous work, [9], we demonstrated in realistic simulations calibrated on real hardware that specific configurations can be obtained using only noisy range and bearing measurements. Building on top of [9], we introduce a novel way to maintain a formation when involved in a leader-tracking task.

Among the literature discussing formation control, the leader-following methods in [8], [10], [11] assume that each robot takes another neighboring robot as a reference point. Here, in Section III, we include in the robotic group a leader that does not participate in the consensus process.

Riccardo Falconi is with the Dipartimento di Elettronica, Informatica e Sistemistica of the Università di Bologna. riccardo.falconi@unibo.it

Sven Gowal and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory of the Ecole Polytechnique Fédérale de Lausanne. svenadrian.gowal@epfl.ch, alcherio.martinoli@epfl.ch

In other words, the followers have to reach the predefined configurations (including the leader) without the consent of the leader.

Continuously maintaining the correct configuration as well as reaching the leader can be considered as two different tasks and giving precedence to one task over the other depending on the current environmental context is very useful. In section III-A, we present a novel approach to dynamically adjust the weights of these tasks, and in Section III-B a first example is introduced focusing on escorting a static leader. Furthermore, as we are interested in systems with a moving leader (that basically plays the role of a moving target), this paper also presents a method based on a nonlinear proportional-integral (PI) controller to actuate the followers (Section III-C). This enables the convergence of the followers to the specified formation as we will see in Section IV. Finally, in Section V, we draw conclusions and suggest future work.

II. BACKGROUND

A. Basic Notions on Graph Theory

In this section the main graph theory results are summarized. An undirected graph with N elements is defined as a pair $G = (V, E)$, where

- $V = \{v_i, i = 1 \ldots N\}$ is the *Vertex set*,
- $E \subseteq V \times V$ is the *Edge set*.

As we deal with undirected graphs, the elements of E are unordered pairs of elements, i.e. $(v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$. A path over G connecting two nodes v_i, v_j is defined as $P_{ij}^G = \{[v_k, v_{k+1}], v_k \neq v_{k+1}, k = i \ldots j, \}$ (we assume, without loss of generality, that $i < j$). A graph G is connected if $\exists P_{ij}^G, \forall v_i, v_j \in V$. The i-th node *Neighbors subset* is defined as $\mathcal{N}_i = \{\forall v_j \in V : (v_i, v_j) \in E\}$, and the *degree* of v_i is defined as $\Delta_i = |\mathcal{N}_i|$.

Given a graph $G = (V, E)$ and an orientation map defined over the edge set, we can define the *incidence matrix* $\mathcal{I} \in \mathbb{R}^{N \times |E|}$ as:

$$\iota_{i,k} = \begin{cases} -1 & \text{if } \varepsilon_k = (n_i, n_j) \\ 1 & \text{if } \varepsilon_k = (n_j, n_i) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $|E|$ is the cardinality of the edge set and ε_k is the k-th edge of G. When the orientation map is not defined, a random orientation can be chosen. An example is reported in Figure 1(a), where four agents are connected by a complete graph with a random orientation. The corresponding
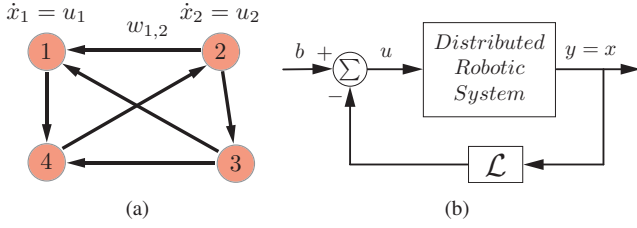
Fig. 1. Two equivalent forms of the consensus algorithm: (a) a network of agents modeled as integrators with weighted and randomly oriented edges and (b) the feedback loop that performs the algorithm on a MIMO system with single integrator agents.

Fig. 2. Kinematic model of a non-holonomic wheeled robot. The red circle identifies the front side.

Incidence Matrix is:

$$\mathcal{I} = \begin{bmatrix} 1 & 1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 1 \end{bmatrix} \quad (2)$$

The definition of the *incidence matrix* allows us to define the *Laplacian matrix* as

$$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^T \quad (3)$$

where the *Weight matrix* $\mathcal{W} \in \mathbb{R}^{|E| \times |E|}$, $\mathcal{W} = \mathrm{diag}\left(\{w_j, \forall \varepsilon_j \in E\}\right)$ is a diagonal matrix that can be used to change the weights assigned to the edges. In particular, if at least one weight differs from 1, the Laplacian matrix is addressed as *Weighted Laplacian matrix*. As explained in [12], the Laplacian matrix of a connected graph has some interesting properties:

a) $eig(\mathcal{L}) = \{0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N\}$

and, in case λ_1 is a simple eigenvalue (i.e. $0 = \lambda_1 < \lambda_2$),

b) $null(\mathcal{L}) = span\{\mathbf{1}\} \Rightarrow \mathcal{L}\mathbf{1} = \mathbf{0}$

where $\mathbf{1} = [1 \ldots 1]^T$ and $\mathbf{0} = [0 \ldots 0]^T$ are vectors of N elements all equal to 1 and 0 respectively.

B. The Consensus Problem

The *consensus problem* [13] is a well-known and widely studied problem in the field of decentralized control. It starts by considering all the agents of a group as holonomic kinematic models:

$$\dot{x}_i = u_i \quad (4)$$

where x_i is the state of the i-th robot. The solution of the consensus problem for N agents, whose goal is to drive the whole system to a final common state, can be solved with the Laplacian based feedback method. The feedback control is in the form

$$\dot{x} = u = -\mathcal{L}x + b \quad (5)$$

In Figure 1 two different representations of the *Laplacian feedback control* are depicted: in Figure 1(a) a random orientation map has been defined over the graph in order to define $\mathcal{L}$ as in Equation 3, in Figure 1(b) a *bias* has been introduced in order to obtain a predefined steady state.

As all the eigenvalues $\lambda_{i,i=2...N}$ of $\mathcal{L}$ are greater than 0, the autonomous closed loop system is marginally stable and the state vector $x = [x_1 \ldots x_N]^T$ converges to the null space
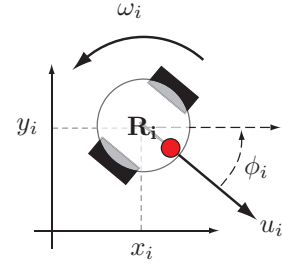
of $\mathcal{L}$, i.e. $x(t) \to \bar{x}$ as $t \to \infty$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$ is the time-independent common final state for single integrator systems. If we have a *bias*, the steady state is $\bar{x} = -\mathcal{L}b$

The convergence ratio of a system based on the Laplacian consensus feedback can be calculated depending on the eigenvalues of $\mathcal{L}$ [12]:

$$\|x(t) - \mathbf{1}\| \leq \|x(0) - \mathbf{1}\|e^{-\lambda_2 t} \quad (6)$$

where λ_2 is the lowest non-zero eigenvalue of $\mathcal{L}$ and $\mathbf{1} = [1 \ldots 1]^T$. It follows that, by changing the values of the matrix $\mathcal{W}$, it is possible to change the convergence by modifying the λ_2 eigenvalue.

C. Consensus with Real Robots

As pointed out in Section II-B, the consensus problem is usually solved on the assumption that the vehicles are modeled as single integrators. In reality, this assumption can not be considered true because robots have kinematic constraints. A typical example of robot is the differential wheeled robot depicted in Figure 2. The kinematic equations of the i-th robot would be:

$$\begin{cases} \dot{x}_i &= u_i \cos(\phi_i) \\ \dot{y}_i &= u_i \sin(\phi_i) \\ \dot{\phi}_i &= \omega_i \end{cases} \quad (7)$$

where u_i is the linear speed, ω_i the rotational speed and the vector $[x_i \ y_i \ \phi_i]^T$ forms the triplet defining the absolute pose. The dynamics of the robot model are ignored due to the very low mass of our hardware platform, the Khepera III robot [14]. Further, we introduce the constraint that all the robots gather relative information provided by a local line-of-sight range and bearing module [15]. This means that often a given robot can not acquire the position and orientation
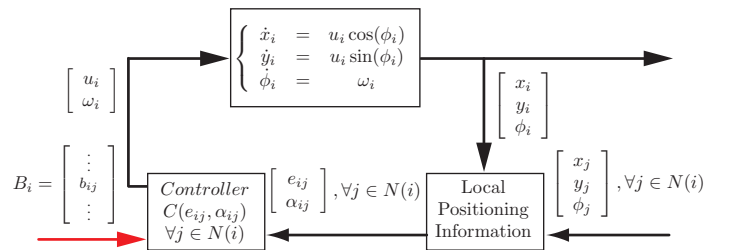

Fig. 3. Feedback scheme for formation keeping using consensus algorithm for a group of differential-wheeled robots with local sensing.

of the teammates because of the distance or occlusions (e.g. when a teammates is in between two others).

In order to consider the nonlinearities induced by the robot model, the control loop is modified as depicted in Figure 3, where the bias input B can be used to achieve a predefined formation. Given a group of N robots, we have shown in [9] that, under the assumption of a connected communication graph, it is possible to drive the system to a predefined steady-state by exploiting only local data and an ad-hoc broadcasting protocol (without that broadcasting protocol, the graph needs to be complete). The proportional controller that stabilized the system for each robot $\mathbf{R_i}$ was:

$$
\begin{aligned}
u_i(t) &= K_u \cdot \bar{e}_i(e_{i,j}(t), \alpha_{i,j}(t)) \cdot \cos(\bar{\alpha}_i(e_{i,j}(t), \alpha_{i,j}(t))) \\
\omega_i(t) &= K_w \cdot \bar{\alpha}_i(e_{i,j}(t), \alpha_{i,j}(t))
\end{aligned}
\tag{8}
$$

where K_u, K_w are two positive constants and $\bar{e}_i(\cdot)$ and $\bar{\alpha}_i(\cdot)$ are

$$
\begin{aligned}
\bar{e}_i(e_{i,j}(t), \alpha_{i,j}(t)) &= \sqrt{\bar{e}_{x,i}^2(t) + \bar{e}_{y,i}^2(t)} \\
\bar{\alpha}_i(e_{i,j}(t), \alpha_{i,j}(t)) &= \operatorname{atan2}(\bar{e}_{y,i}(t), \bar{e}_{x,i}(t))
\end{aligned}
$$

with

$$
\begin{aligned}
\bar{e}_{x,i}(t) &= \sum_{j=1}^{\Delta_i} \left[-\mathcal{L}_{i,j} \cdot e_{i,j}(t) \cdot \cos(\alpha_{i,j}(t))\right] \\
\bar{e}_{y,i}(t) &= \sum_{j=1}^{\Delta_i} \left[-\mathcal{L}_{i,j} \cdot e_{i,j}(t) \cdot \sin(\alpha_{i,j}(t))\right] \quad .
\end{aligned}
$$

Hence, the control is a function of the acquired data: $e_{i,j}(t)$ is the Euclidean distance between the robot $\mathbf{R_i}$ and the robot $\mathbf{R_j}$, and $\alpha_{ij}(t)$ is the azimuth of $\mathbf{R_j}$ with respect to $\mathbf{R_i}$ (see Figure 4). An extended demonstration of the controller stability can be found in [17]. In Figure 5 four robots start in random positions and converge to a square formation with radius $r = 2.0$ m.

III. GRAPH-BASED ESCORTING MISSION

A. Graph-based Behavioral Control

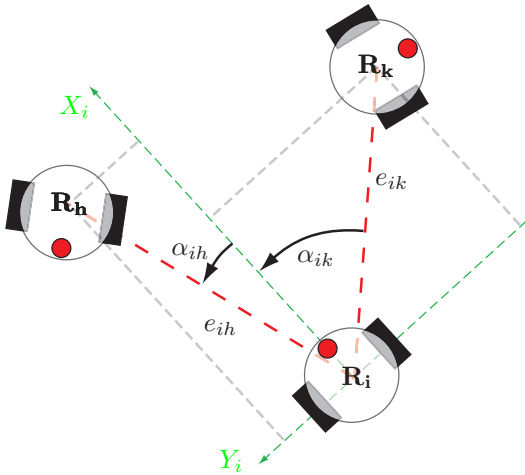Let's suppose we have a group of N robots whose communication graph is complete. The definition of the Laplacian

Fig. 4. Definition of the local measurement data with respect to the robot $\mathbf{R_i}$.
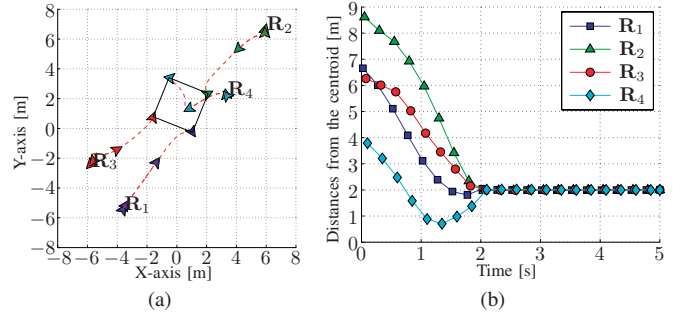

(a) (b)

Fig. 5. Evolution of a four-differential-wheeled-robot system converging to a square formation: (a) trajectories of the robots and (b) distances to the center of mass.

matrix as reported in Equation 3 allows us to define a *Weight matrix* that can be used to change how different edges can affect the dynamics of the system. Our goal is to understand how these changes should occur when one or more leaders are introduced in the graph in order to get a predefined group behavior.

In the general case where many *leader nodes* are introduced in the group, the graph G become $\overline{G} = (\overline{V}, \overline{E})$. The *vertex set* $\overline{V}$ holds $\overline{V} = V_f \bigcup V_l$ and $V_f \bigcap V_l = 0$, where V_f is the *followers subset* and V_l is the *leaders subset*. This partition of the *vertex set* was already presented in [18], where authors focused on the possibility of using leaders to drive the followers to a predefined configuration.

Our idea starts from the possibility of exploiting the partition of the *edge set* that derives from the follower/leader partition of the *vertex set*. The *edge set* can be partitioned in three subsets, that are:

- $E_f \subseteq V_f \times V_f$ (inter-follower edges)
- $E_l \subseteq V_l \times V_l$ (inter-leader edges)
- $E_{fl} = E_{lf} \subseteq V_f \times V_l$ (leaders-to-follower edges)

We consider the leaders as a part of the group, the columns of the *Incidence matrix* can be rearranged as:

$$
\mathcal{I} = \left[\begin{array}{c|c|c} \mathcal{I}_{ff} & \mathcal{I}_{fl} & 0 \\ \hline 0 & \mathcal{I}_{lf} & \mathcal{I}_{ll} \end{array} \right]
\tag{9}
$$

where $\mathcal{I}_{ff}$ is the incidence matrix corresponding to the inter-followers edges, $\mathcal{I}_{fl}$, $\mathcal{I}_{lf}$ contain the tails and the heads of the edges between followers and leaders respectively, and $\mathcal{I}_{ll}$ corresponds to the inter-leaders edges. We can partition the *Weight matrix* as

$$
\mathcal{W} = \begin{bmatrix} \mathcal{W}_{ff} & 0 & 0 \\ 0 & \mathcal{W}_{fl} & 0 \\ 0 & 0 & \mathcal{W}_{ll} \end{bmatrix}
\tag{10}
$$

where $\mathcal{W}_{ff} = \operatorname{diag}\{w_j, \forall \varepsilon_j \in E_f\}$, $\mathcal{W}_{fl} = \operatorname{diag}\{w_j, \forall \varepsilon_j \in E_{fl}\}$ and $\mathcal{W}_{ll} = \operatorname{diag}\{w_j, \forall \varepsilon_j \in E_l\}$. Roughly speaking, $\mathcal{W}_{ff}$ is a diagonal matrix that collects the weights of the inter-follower edges, $\mathcal{W}_{fl}$ is a diagonal matrix that collects the weights of the follower-to-leader edges and $\mathcal{W}_{ll}$ is a diagonal matrix that collects the weights of the inter-leader edges. Given these considerations, it follows that the *Weighted Laplacian matrix* defined in
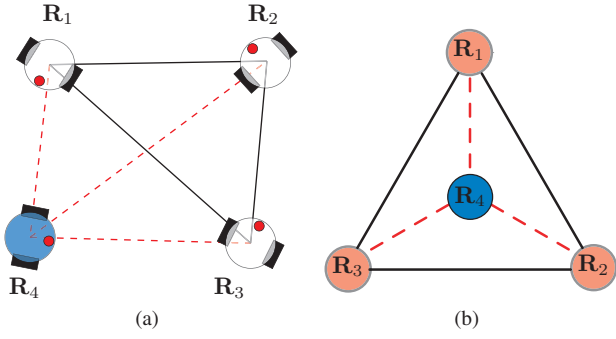
Fig. 6. A leader node ($\mathbf{R_4}$) is introduced into a complete graph with new edges (dotted lines). Figure (b) shows our desired final configuration.
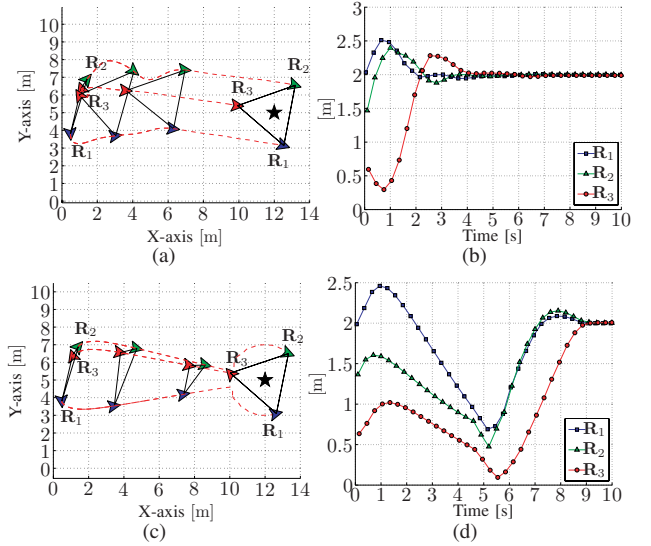

Fig. 7. Trajectories and distances to the center of mass of a group of followers while they converge around the leader using (a)-(b) $w_{ff} = 10, w_{fl} = 1$ and (c)-(d) $w_{ff} = 1, w_{fl} = 10$. The leader vehicle is represented by a black star at $x = 12m, y = 5m$.

Equation 3 can be partitioned in order to consider follower-leader subgroups and corresponding weights. The Laplacian matrix becomes:

$$\overline{\mathcal{L}}_{\mathcal{W}} = \left[\begin{array}{c|c} \overline{\mathcal{L}}_{ff} & \overline{\mathcal{L}}_{fl} \\ \hline \overline{\mathcal{L}}_{fl}^{T} & \overline{\mathcal{L}}_{ll} \end{array} \right] \tag{11}$$

where:

- $\overline{\mathcal{L}}_{ff} = \mathcal{I}_{ff}\mathcal{W}_{ff}\mathcal{I}_{ff}^{T} + \mathcal{I}_{fl}\mathcal{W}_{fl}\mathcal{I}_{fl}^{T}$
- $\overline{\mathcal{L}}_{fl} = \mathcal{I}_{fl}\mathcal{W}_{fl}\mathcal{I}_{lf}^{T}$
- $\overline{\mathcal{L}}_{ll} = \mathcal{I}_{ll}\mathcal{W}_{ll}\mathcal{I}_{ll}^{T} + \mathcal{I}_{lf}\mathcal{W}_{fl}\mathcal{I}_{lf}^{T}$

We observe that different weights can affect how the system converges to the leaders position (i.e., as the leaders are not affected by the followers, they can be considered as an anchor point for the system).

B. Escorting a Static Leader

As we are investigating how we can obtain a desired behavior for a graph with one leader, that is $|E_l| = 0$, the matrices in Equations 9,10 become:

$$\mathcal{I} = \left[\begin{array}{c|c} \mathcal{I}_{ff} & \mathcal{I}_{fl} \\ \hline 0 & \mathcal{I}_{lf} \end{array} \right] \quad \mathcal{W} = \left[\begin{array}{cc} \mathcal{W}_{ff} & 0 \\ 0 & \mathcal{W}_{fl} \end{array} \right] \tag{12}$$

and $\overline{\mathcal{L}}_{ll} = \mathcal{I}_{lf}\mathcal{W}_{fl}\mathcal{I}_{lf}^{T}$. As an example, let us consider the completely connected system depicted in Figure 6, directly derived from the example depicted in Figure 1(a) and where robot $\mathbf{R_4}$ is considered as the leader. The incidence matrix in Equation 2 can be rewritten as:

$$\mathcal{I} = \left[\begin{array}{ccc|ccc} -1 & -1 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 1 & 1 & 1 \end{array} \right] \tag{13}$$

where the divisions correspond to Equation 12. As the *Weight matrix*, we define:

$$\mathcal{W} = \text{diag}\left\{ w_{ff}, \ w_{ff}, \ w_{ff}, \ w_{fl}, \ w_{fl}, \ w_{fl} \right\} \tag{14}$$

with $w_{ff}, w_{fl} > 0$. From Equation 11, it follows that:

$$\overline{\mathcal{L}}_{\mathcal{W}} = \left[\begin{array}{cccc} 2w_{ff} + w_{fl} & -w_{ff} & -w_{ff} & -w_{fl} \\ -w_{ff} & 2w_{ff} + w_{fl} & -w_{ff} & -w_{fl} \\ -w_{ff} & -w_{ff} & 2w_{ff} + w_{fl} & -w_{fl} \\ -w_{fl} & -w_{fl} & -w_{fl} & 3w_{fl} \end{array} \right] \tag{15}$$

with its eigenvalues,

$$\text{eig}(\overline{\mathcal{L}}_{\mathcal{W}}) = [0, \ \lambda_2, \ \lambda_3, \ \lambda_4]^{T}, \tag{16}$$

depending only on the values of w_{ff}, w_{fl}. As pointed out in Equation 6, the lower non-zero eigenvalue of the *Laplacian matrix* defines the upper bound of the convergence time. In our example this eigenvalue λ_2 is:

$$\lambda_2 = \begin{cases} 4w_{ff} = 4w_{fl} & \text{if} \quad 3w_{ff} + w_{fl} = 4w_{fl} \\ 3w_{ff} + w_{fl} & \text{if} \quad 3w_{ff} + w_{fl} < 4w_{fl} \\ 4w_{fl} & \text{if} \quad 3w_{ff} + w_{fl} > 4w_{fl} \end{cases} \tag{17}$$

Intuitively, it means that if $w_{ff} < w_{fl}$ the system converges to the leader position before the followers converge to their formation; on the other hand, if $w_{ff} > w_{fl}$ the followers converge to their formation before reaching the leader position. Both behaviors present positive and negative aspects: if $w_{ff} \gg w_{fl}$, the followers are strongly bounded to stay in formation and, in case of obstacles in the environment, they may not be able to reach the leader; if $w_{ff} \ll w_{fl}$, the followers are not forced to preserve their formation but they surround the target only when they are close to it. The behavior of a group of followers converging to the leader position and surrounding it with a regular formation with a diameter of 2m is depicted in Figure 7 using different weights.

To account for the above issues, we can define $w_{ff,i}(\cdot)$ and $w_{fl,i}(\cdot)$ as functions depending on the distance d between each follower and the leader. For our purposes, we use the sigmoid functions $w_{ff,i}(d), \ \forall \varepsilon_i \in E_f$ and $w_{fl,j}(d), \ \forall \ \varepsilon_j \in E_{fl}$ defined as:

$$\begin{aligned} w_{ff,i}(d) &= \beta_1 \left(1 - \frac{1}{1 + e^{-s(d-d_1)}} \right) + \delta_1 \\ w_{fl,j}(d) &= \beta_2 \left(\frac{1}{1 + e^{-s(d-d_2)}} \right) + \delta_2 \end{aligned} \tag{18}$$
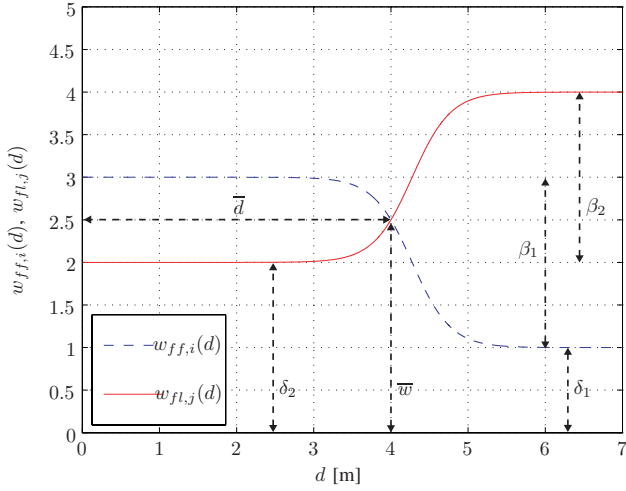
Fig. 8. Sigmoid functions used to change the behavior of the followers depending on the distance from the leader.
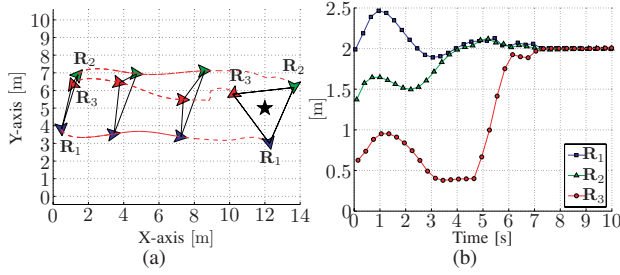

Fig. 9. Evolution of a group of three followers while they converge to the leader position using weights defined in Equation 18: (a) trajectories of the robots and (b) distances to the center of mass of the followers. The leader vehicle is represented by a black star at $x = 12m, y = 5m$.

where s is the slope of the sigmoid functions, β_1, β_2 are used to define the excursion of each sigmoid, δ_1, δ_2 are used to set the minimum value of each function and d_1, d_2 are used to the define the sigmoid functions' crossing point. As $w_{ff,i}(d)$, $w_{fl,j}(d)$ are used by the system to assign different priorities to the formation keeping and the leader surrounding task, a threshold distance $\overline{d}$ can be defined where $w_{ff,i}(\overline{d}) = w_{fl,j}(\overline{d}) = \overline{w}$. It means that when $d = \overline{d}$, both behavioral functions have the same magnitude. The two parameters d_1, d_2 become:

$$d_1(\overline{d}, \overline{w}, \beta_1, \delta_1) = \overline{d} + \frac{1}{s} \ln \left(\frac{\beta_1}{\beta_1 - \overline{w} + \delta_1} - 1 \right)$$

$$d_2(\overline{d}, \overline{w}, \beta_2, \delta_2) = \overline{d} + \frac{1}{s} \ln \left(\frac{\beta_2}{\overline{w} - \delta_2} - 1 \right)$$

In Figure 8 two sigmoid functions are depicted with their parameters. In Figure 9 the behavior of a group of three followers is shown with the parameters $s = 4$, $\beta_1 = 2$, $\beta_2 = 2$, $\delta_1 = 1$, $\delta_2 = 2$, $\overline{w} = 2.5$, $\overline{d} = 2r$, where r is the desired radius for the final formation.

Note that in this case, the followers converge to the leader position *while* they achieve the regular configuration. Intuitively, it is easy to understand the power of this approach: once some key points are defined (i.e. a threshold
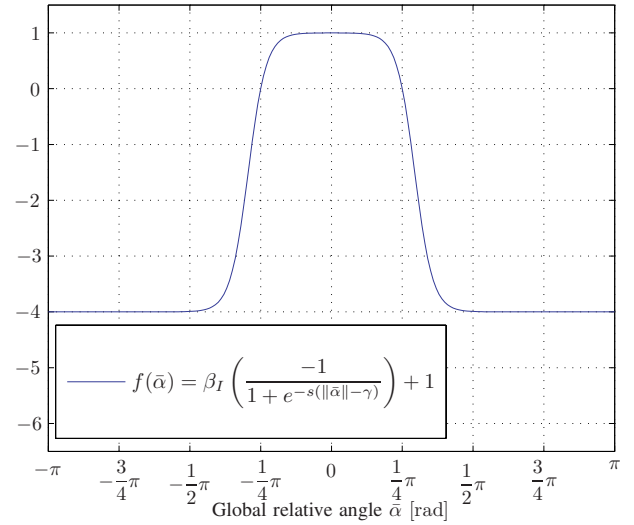

Fig. 10. Function used to modulate the integral part of the controller depending on the global relative angle $\bar{\alpha}$.

distance), the behavior of the group can change as smoothly as desired simply by changing the parameters of the functions $w_{ff,i}(d)$, $w_{fl,j}(d)$, and different behaviors can be defined for the *followers subgroup*, for the *leaders subgroup* and for their interconnections.

C. Escorting a Moving Leader

Let's suppose now that the leader robot is not static. The leader can be a vehicle with a different kinematic model with respect to the followers, but we will assume that here it moves along straight segments. This assumption is not limiting the generality of our approach and as usual the leader is a vehicle with a complete knowledge of the environment where the following group is moving and it can calculate a trajectory to satisfy its constraints. We note that the leader trajectory is not a priori known by the followers. As an example, one could consider that the followers are Unmanned Ground Vehicles (UGV) and the leader is an Unmanned Aerial Vehicle (UAV). With a moving leader, the proportional control in Equation 8 is not suitable anymore to maintain the formation around the leader. The main idea is to consider that when the followers are aligned with the leader (i.e. they are moving in the same direction), the kinematics of the followers can be simplified to a single integrator. This means that to reach the leader, a *integral* part must be added to the forward control u .

At this point, a new problem arises: since the distances are acquired using only local information, we have that $\bar{e}(e_{i,j}, \alpha_{i,j}) \geq 0$ and, thus, the integral part of the controller cannot be discharged. To solve this issue, we modify the controller as follows:

$$\begin{cases} u_i(t) = K_u \cdot \bar{e}_i \cdot \cos(\bar{\alpha}_i) + K_I \cdot \int_0^t f(\bar{\alpha}_i) \cdot \bar{e}_i \, dt \\ \omega_i(t) = K_w \cdot \bar{\alpha}_i \end{cases}$$

$$(19)$$

where K_u, K_w, K_I are positive constants and $f(\bar{\alpha})$ is a nonlinear function used to modulate the value of the integral
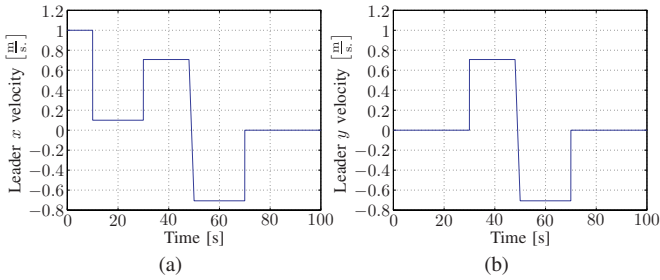
Fig. 11. Velocity profile of the leader considered as a massless point: (a) profile on the x-axis and (b) on the y-axis.
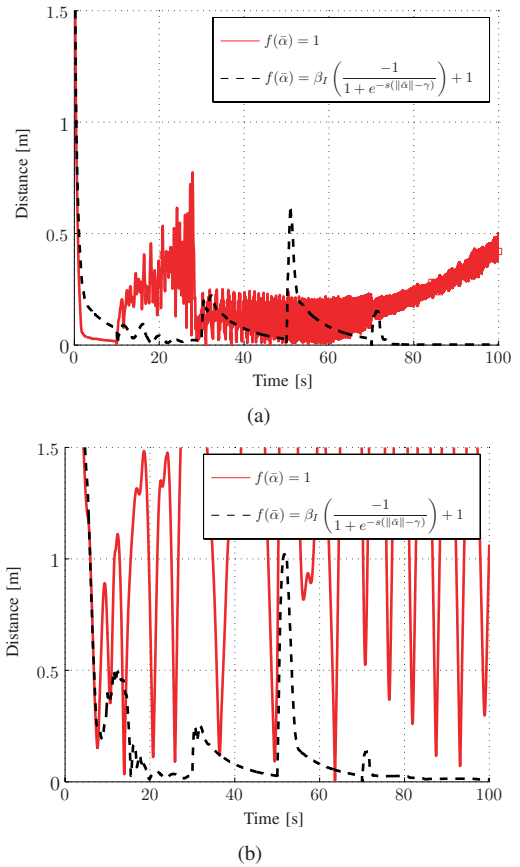

Fig. 12. Distance between the center of mass of the followers subgroup and the leader position (a) without motor saturation and (b) with motor saturation. When $f(\bar{\alpha}) = 1$ the system is unstable.

part. The function $f(\bar{\alpha})$ has to be chosen to ensure that if the followers are aligned with the leader (i.e. $\bar{\alpha} = 0$) the integral action allows them to reach it, otherwise (i.e. $\bar{\alpha} \neq 0$) $f(\bar{\alpha})$ has to be negative so that the followers can move slowly and align to the velocity of the leader. The function we choose is the sigmoid function

$$f(\bar{\alpha}) = \beta_I \left(\frac{-1}{1 + e^{-s(\|\bar{\alpha}\| - \gamma)}} \right) + 1; \qquad (20)$$

where β_I is amplitude of the function and s is the slope. The parameter γ is defined by choosing the angle $\bar{\alpha}_0$ such that $f(\bar{\alpha}_0) = 0 : \gamma = \|\bar{\alpha}_0\| + \frac{1}{s} \ln(\beta_I - 1)$. In Figure 10 the *modulation function* $f(\bar{\alpha})$ with $\beta_I = 5$, $s = 10$ and $\gamma = 0.924$ (i.e. $\bar{\alpha}_0 = \frac{\pi}{4}$) is depicted. As an example, let us consider the system depicted in Figure 6, where the leader robot is moving. To be as general as possible, we supposed the leader can move without kinematic constraints (i.e. it can be considered as a massless point). The trajectory of the leader on the plane can be described with velocity profiles on the x and y axes. The leader velocity profile is depicted in Figure 11, while in Figure 12 the distance between the center of mass of the followers and the leader is depicted in case the *modulation function* is used (dotted line) or when $f(\bar{\alpha})$ is constant equal to one (solid line). The same simulations are performed when the motor speed is limited to 2 m/s (12(b)) or not (12(a)). In both cases, the leader and the followers start at the same positions. Looking at these pictures, some considerations can be pointed out: in the first part of the simulation ($0 < t \leq 10$[s.]) the behavior of the follower group is almost the same, due to the fact that the leader is moving on a line with constant velocity; at $t > 10$[s.], after the first change of trajectory, the behaviors diverge because, in case of $f(\bar{\alpha}) = 1$, the followers see the error $\bar{e}$ increasing, the controller continues integrating and the system becomes instable.

IV. EXPERIMENTS

Experiments were performed using Khepera III robots [14], developed by K-Team in collaboration with the Distributed Intelligent Systems and Algorithms Laboratory (DISAL) at EPFL. This robot has a diameter of 12 cm, making it appropriate for multi-robot indoor experiments. The goal of the three follower robots, whose initial orientation is random, is to match their mean position with the leader and to reconfigure in a regular

configuration of 0.5 m diameter. The controller of the robots has $K_u = 35000$, $K_w = 50000$ and $K_I = 2000$ (the rest of the parameters are the same as used for Figure 12(b)), and a data broadcasting has been implemented to avoid instability problems due to line-of-sight occlusions, as already explained in [9]. A Braitenberg [19] controller has been added on top of the formation control to avoid obstacles.

A. Range and Bearing

A hardware extension board for the Khepera III robot has been developed in [15] to enable robots to find their relative positions. Figure 13(a) shows the sixteen evenly-spaced infrared Light Emitting Diodes (LEDs) that this module uses. This range and bearing board has also the ability to broadcast low bit rate communication packets using the IR emitters.

B. Simulated Robots

1) Experimental Setup: Experiments are conducted in Webots [16], a realistic mobile robotic simulator carefully calibrated using real robotic data. All the sensors and actuators of the simulated robotic platform were calibrated to match reality: in particular, a slip noise has been added to the wheels, infrared sensors are modeled based on data
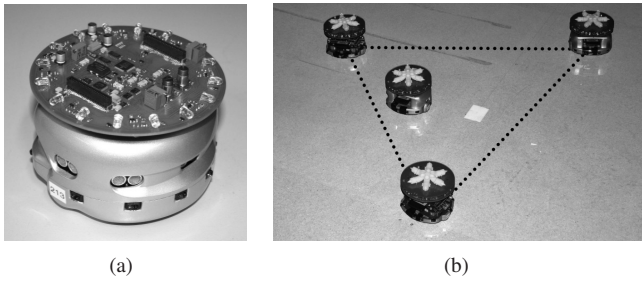
Fig. 13. The range and bearing board developed at EPFL on a Khepera III robot 13(a) and four robots performing the escorting algorithm 13(b).

Fig. 15. Average and standard deviation of the mean square error of the distances between each robot and the desired distances depending on time and with obstacles.

Fig. 16. Top view of the reported positions of the robots in the real arena during a run. Followers are red dots, the leader is a black star. The run starts with the robots placed on the left side and finishes when they reach the right side.

sheets, and the range and bearing platform is affected by a noise of 10% in the estimation of the distance and a 0.1 radian noise in the relative angle. Two sets of experiments are performed in a 15×15 m arena, one without obstacles and one with. Obstacles are represented by 1 m diameter cylinders randomly placed in the environment around the initial position of the robots. In both sets of experiments, the four robots are randomly placed in 3×3 m area. The leader initially moves straight at 11 cm/s (approximately one robot size per second), after two minutes it starts to move along a circle of radius 1 m at 7 cm/s for 30 seconds, it then resumes its prior movement and stops moving after another 1 minute and 30 seconds.

2) Results: The position of each robot is monitored during a run. After 100 runs, the average mean square error (MSE) between the actual distances between each pair of robot and the desired distances is computed. Figure 14 shows the MSE without obstacles when data broadcasting is enabled (*Full broadcast*) and disabled (*No broadcast*). In both cases, convergence of the formation (and this includes the leader) is quickly achieved (after 50 seconds in average) as the leader moves at 11 cm/s and we limit the maximal speed of the simulated Khepera III to 26 cm/s. We note also that the control is reactive to changes in the leader movement.
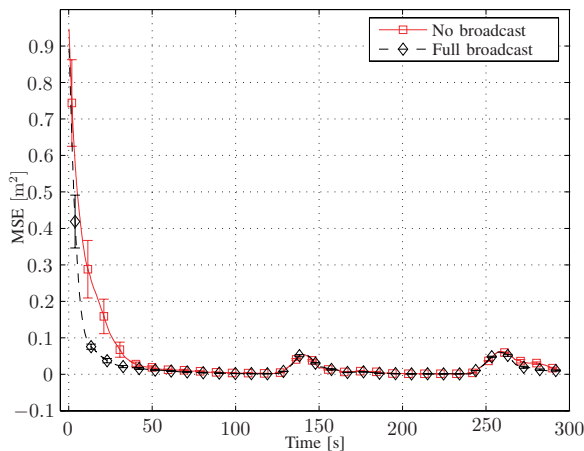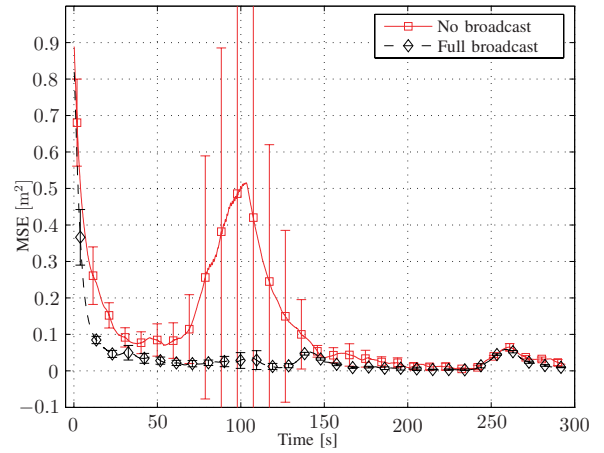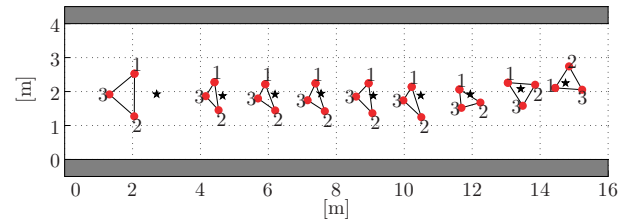
Figure 15 shows the MSE with obstacles. In average the first obstacle is hit by the leader at around 50 seconds and we observe that with broadcasting enabled, the controller stays reactive and is easily able to reach the target and keep the formation intact. Without broadcasting the system looks unstable but as soon as the target stops for 30 seconds the followers are able to come back.

Overall, the simulation results show the very good performance of our approach.

C. Real Robots

1) Experimental Setup: To test our algorithm on real robots, we have chosen the biggest arena available at our lab, that is a 16×4 meters arena. Khepera III robots equipped with range and bearing boards (see Figure 13(a)) are initially placed in the arena as shown on the left part of Figure 16. The leader (black star) is positioned in front of the followers (red dots) facing the right-end of the arena.

To push the limits of our approach on actual hardware, the leader is set to move at 19 cm/s, the update frequency of the positioning board is set to 5 Hz and data broadcasting is enabled.

2) Results: The position of each robot is monitored with SwisTrack [20], an open-source tracking software. After around 65 seconds, the leader reaches the end of the arena and the experimental run is stopped. After 12 runs, the

Fig. 14. Average and standard deviation of the mean square error of the distances between each robot and the desired distances depending on time and without obstacles.
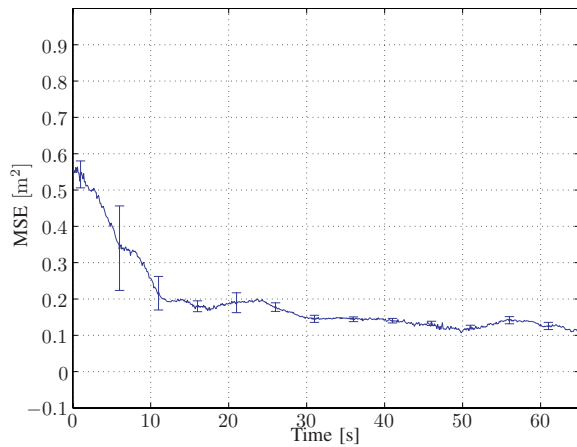
Fig. 17. Average and standard deviation of the mean square error of the distances between each robot and the desired distances depending on time in the real arena.

average MSE is computed and shown in Figure 17. We notice that although the MSE does not reach zero, it stabilizes after only 30 seconds at 0.13 m^2. This corresponds to an average error of 15 cm (about one robot size) on each link. This result not only confirms the good performance of our approach but also its ability to stabilize under difficult conditions (i.e. a fast moving leader).

V. CONCLUSION

In this paper, we demonstrated that we could drive non-holonomic robots to a specific formation while following a leader robot using only noisy local positioning information. We derived a nonlinear PI controller to enable the formation of followers to reach, surround and escort the moving leader. We showed that different behaviors can be achieved by changing the weights on the edges of the communication graph. We also tested the robustness of our control under challenging conditions such as obstacle field arenas and unpredictable leader trajectories. The next step of this work will focus on the possibility to add more complex behaviors using the same framework.

ADDITIONAL MATERIAL

A video showing four simulated Khepera III robots is available on `http://www5.epfl.ch/swis/page35885.html`.

REFERENCES

[1] M. Egerstedt. *Graph-Theoretic Methods for Multi-Agent Coordination*, ROBOMAT, 2007.
[2] K. Kyriakopoulos, D. Dimarogonas, *On the state agreement problem for multiple unicycles with varying communication links*, IEEE Conf. on Decision and Control, pp. 4283-4288, 2006.
[3] K. Kyriakopoulos, D. Dimarogonas, *On the rendezvous problem for multiple nonholonomic agents*, IEEE Trans. on Automatic Control, Vol. 52, No. 5, pp. 916-922, 2007.
[4] B. Smith, J. Wang, M. Egerstedt and A. Howard, *Automatic Formation Deployment of Decentralized Heterogeneous Multi-Robot Networks with Limited Sensing Capabilities*, IEEE Int. Conf. on Robotics and Automation, pp. 730-735, 2009.
[5] B. Smith, J. Wang, and M. Egerstedt, *Persistent formation control of multi-robot networks*, in Proc. of the IEEE Conf. on Decision and Control, pp. 471-476, 2008.
[6] M. Ji and M. Egerstedt, *Distributed Coordination Control of Multiagent Systems while Preserving Connectedness*, IEEE Trans. on Robotics, Vol. 23, No. 4, pp. 693-703, 2007.
[7] F. Zhang and S. Haq, *Boundary following by robot formations without gps*, Proc. of the IEEE Int. on Robotics and Automation, pp. 152-157, 2008.
[8] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer and C. Taylor, *A Vision-Based Formation Control Framework*, IEEE Trans. on Robotics and Automation, Vol. 18, No. 5, pp. 813-825, 2002.
[9] R. Falconi, S. Gowal, J. Pugh, A. Martinoli, *Graph-Based Distributed Control for Non-holonomic Vehicles Engaged in a Reconfiguration Task using Local Positioning Information*, DOI:10.41081ICST, Robocomm, 2009.
[10] J. Fredslund and M. Mataric, *A general algorithm for robot formations using local sensing and minimal communication*, IEEE Trans. on Robotics and Automation, Vol. 18, No. 5, pp.837-846, 2002.
[11] J. Shao, X. Guangming and L. Wang, *Leader-following Formation Control of Multiple Mobile Robots*, IEEE Int. Symp. on Intelligent Control, pp. 808-813, 2005.
[12] C. Godsil, G. Royle, *Algebraic Graph Theory*, Springer, 2001.
[13] R. Olfati-Saber, J. A. Fax, R. M. Murray, *Consensus and Cooperation in Networked Multi-Agent Systems*, Proc. of the IEEE, Vol. 95, No. 1, pp 215-233, 2007.
[14] http://www.k-team.com
[15] J. Pugh, X. Raemy, C. Favre, R. Falconi and A. Martinoli, *A Fast On-Board Relative Positioning Module for Multi-Robot Systems*, IEEE Trans. on Mechatronics, focus section on Mechatronics and Multi-robot systems, Vol. 14, pp. 988-1001, 2009.
[16] O. Michel, *Webots: Professional mobile robot simulation*, Journal of Advanced Robotic Systems, Vol.1, No.1, pp. 39-42, 2004.
[17] R. Falconi, *Coordinated Control of Robotic Swarms in Unknown Environments*, Ph.D. dissertation, Università di Bologna, 2009.
[18] M. Ji, A. Muhammad and M. Egerstedt, *Leader-Based Multi-Agent Coordination: Controllability and Optimal Control*, IEEE American Control Conf., pp. 1358-1363, 2006.
[19] V. Braitenberg, *Vehicles, Experiments in Synthetic Psychology*, The MIT Press, 1984.
[20] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot and A. Martinoli, *SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems*, IEEE/RSJ 2008 Int. Conf. on Intelligent Robots and Systems, pp. 4004-4010, 2008.