The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

The Null-Space Based Behavioral Control for a Team of Cooperative Mobile Robots with Actuator Saturations

Filippo Arrichiello Stefano Chiaverini Giovanni Indiveri Paola Pedone

Abstract— **This paper presents the application of the Null-Space based Behavioral (NSB) approach to the motion control of a team of mobile robots with velocity saturated actuators. In particular, the proposed solution aims at managing actuator velocity saturations by dynamically scaling task velocity commands so that the hierarchy of task priorities is preserved in spite of actuator velocity saturations. The approach is tested on a specific case study where the NSB elaborates the motion directives for a team of six mobile robots that has to entrap and to escort a target. The approach is validated by numerical simulations and by experimental results.**

I. INTRODUCTION

Mobile robots have been object of widespread research in the last decades. Their applications span over service, industrial, military and civil fields and involve missions like exploration, transportation and mobile manipulation. With reference to motion control problems, widely different methods and techniques have been presented in the literature including behavior-based approaches. Among the behavioral approaches, seminal works are reported in the papers [10] and [6], while a comprehensive state of the art is presented in [7]. Behavioral approaches have been also applied to the formation control of multi-robot systems as in, e.g., [17], [15] and [9].

Extending the idea of inverse kinematics techniques for industrial manipulators to the case of mobile robots, a new behavior based approach, namely the Null-Space-based Behavioral control (NSB), has been recently presented in the literature [3]. In particular, the NSB approach is based on the idea of task based kinematical control for industrial manipulators presented in [14], [16], [18] of exploiting eventual kinematical redundancy to try to accomplish more than one motion task simultaneously. Nevertheless, as discussed in [11], in the case of conflicting tasks it is necessary to devise singularity-robust algorithms that ensure proper functioning of the inverse velocity mapping. Based on these works, this idea is developed in [5] in the framework of the singularity-robust task-priority inverse kinematics [11]. The NSB has been introduced in comparison with the main behavior-based approaches in [3] and it has been recently experimentally applied in a large number of multi-robot missions such as formation control or escort/entrap an autonomous target [2].

An important implementation issue that has been often overlooked at when designing kinematics based control laws is related to actuator velocity saturation. Within a kinematics tasked based control architecture, should a low priority task command induce even a single actuator to saturate its velocity, this could irremediably corrupt the high priority task: on the other hand, the common work-around to scale all the actuator speeds to avoid saturations when the original command is too high has the drawback that so doing higher priority tasks are slowed down due to lower priority ones hence somehow violating the very basic idea of priority hierarchy. To avoid these limitations, the NSB solution can be extended with velocity saturation management techniques [12] as described in [8]. The solution for the management of actuator velocity saturations presented in [12] and adopted in [8], although effective, can be over conservative: in particular, to avoid saturations, task commands are scaled down on the basis of a worst-case condition that is not necessarily always met. In this paper, the velocity saturation management technique proposed in [12] is extended to overcome this limit. The new approach, although more complex form an algorithmic point of view, is never over-conservative and still simple enough to be implemented on line. Such solution is then applied to the NSB scenario for a team of robots rather than to a single one as in [8]. The proposed solution is tested, both in simulation and experimentally, on the specific case of the motion control of a team of six mobile robots that has to execute the mission presented in [2] consisting in entrapping a moving target. The adopted experimental platform is a team of Khepera II mobile robots.

II. THE NULL-SPACE BASED BEHAVIORAL CONTROL

The Null-Space based Behavioral control [3] [2] is a behavior-based technique aiming at controlling the motion of robotics systems. Following the main behavior-based approaches and similarly to task-based kinematic control approaches, the NSB approach builds on a decomposition of the overall mission of the team in elementary sub-problems (or *tasks*) that have to be simultaneously managed. More specifically, a task variable $\boldsymbol{\sigma} \in \mathbb{R}^m$ is defined to be controlled . Denoting with $\boldsymbol{p} \in \mathbb{R}^s$ the system configuration, the task variable $\boldsymbol{\sigma}$ is designed so that the mapping $\boldsymbol{\sigma} = \mathbf{f}(\boldsymbol{p})$ is differentiable implying

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \mathbf{f}(\boldsymbol{p})}{\partial \boldsymbol{p}} \mathbf{v} = \mathbf{J}(\boldsymbol{p}) \mathbf{v}, \qquad (1)$$

Authors are listed in alphabetical order.

F. Arrichiello and S. Chiaverini are with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università degli Studi di Cassino, Via G. Di Biasio 43, 03043, Cassino (FR), Italy, {f.arrichiello,chiaverini}@unicas.it, http://webuser.unicas.it/lai/robotica.

G. Indiveri and P. Pedone are with the Dipartimento Ingegneria dell'Innovazione, Università del Salento, via Monteroni, 73100 Lecce, Italy, {giovanni.indiveri, paola.pedone}@unile.it, http://cor.unile.it.

where $\mathbf{J} \in \mathbb{R}^{m \times s}$ is the configuration-dependent task Jacobian matrix and $\mathbf{v} \in \mathbb{R}^s$ is the system velocity. Notice that, in case of a team of l planar robots where $\boldsymbol{p}_i \in \mathbb{R}^2$ is the position of the i^{th} robot, then $\boldsymbol{p} = [\boldsymbol{p}_1^T \dots \boldsymbol{p}_s^T]^T$ and $s = 2\,l$.

For each task, the velocity reference for the robot is elaborated, starting from desired values $\boldsymbol{\sigma}_d$ of the task function, solving the inverse kinematic problem at a differential level. In particular, the NSB makes use of the pseudo-inverse Jacobian of the task function. Thus, the velocity reference of the generic k^{th} task can be calculated as

$$\mathbf{v}_k = \mathbf{J}^{\dagger}\left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda}\widetilde{\boldsymbol{\sigma}}\right), \tag{2}$$

where $\mathbf{J}^{\dagger} = \mathbf{J}^{\mathrm{T}}\left(\mathbf{J}\mathbf{J}^{\mathrm{T}}\right)^{-1}$ (when $\mathbf{J}(\boldsymbol{p})$ is full rank), $\boldsymbol{\Lambda}$ is a suitable constant positive-definite matrix of gains and $\widetilde{\boldsymbol{\sigma}}$ is the task error defined as $\widetilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$.

When the mission is composed of multiple tasks, the overall vehicle velocity is obtained by properly merging the outputs of the individual tasks. If the subscript k of eq. (2) also denotes the priority of the task with, e.g., task 1 being the highest-priority one, the overall robot velocity is elaborated according to [11] as:

$$\mathbf{v}_d = \mathbf{v}_1 + \left(\boldsymbol{I} - \mathbf{J}_1^{\dagger}\mathbf{J}_1\right)\left[\mathbf{v}_2 + \left(\boldsymbol{I} - \mathbf{J}_2^{\dagger}\mathbf{J}_2\right)\mathbf{v}_3\right], \tag{3}$$

where the $\left(\boldsymbol{I} - \mathbf{J}_k^{\dagger}\mathbf{J}_k\right)$ operator represents the null-space projector of the k^{th}-task. Therefore, the generalization of eq. (3) can be written in the form:

$$\mathbf{v}_d = \sum_{k=1}^{N_{task}} \bar{\mathbf{v}}_k\,, \tag{4}$$

being

$$\bar{\mathbf{v}}_k = \begin{cases} \mathbf{v}_1 & \text{if } k = 1 \\ \left[\prod_{i=2}^{k}\left(\boldsymbol{I} - \mathbf{J}_{i-1}^{\dagger}\mathbf{J}_{i-1}\right)\right]\mathbf{v}_k & \text{if } k > 1. \end{cases}$$

It is worth noticing that to guarantee the mission stability the tasks have to verify the properties defined in [4].

III. The saturation management technique

The above described procedure guarantees the compatibility among tasks in the assumption that the overall velocity given by eq. (3) does not exceed in norm the maximum value, say $v_{\max}$, that is physically realizable. Indeed hardware and energy limitations imply that the maximum possible velocity should be bounded. If the commanded velocity $\mathbf{v}_d$, due to the presence of lower priority tasks, should be such that $\|\mathbf{v}_d\| > v_{\max}$, the actual speed of the vehicle relative to a $\mathbf{v}_d$ command would result in $v_{\max}\,\mathbf{v}_d/\|\mathbf{v}_d\|$: in such case, in spite of the null-space projection technique, lower priority tasks would actually still conflict with higher priority ones at least in terms of task error convergence rate. To overcome this issue, task velocity commands should be normalized in norm so that lower priority commands do not conflict with higher priority ones due to saturation effects. The solution discussed in [8] and [12], that is not here reported for brevity, has the following shortcomings:

1 There is only one saturation threshold (or capacity) for each task rather than for each actuator. This is a limit because, in general, a system may have actuators with different saturation thresholds.
2 The saturation is symmetric, namely the lower and upper limit are equal in norm and opposite in sign. This does not allow to cope with actuators having asymmetrical saturation thresholds.
3 The decision on whether to scale a lower priority velocity command or not is taken only based on the infinity norm of the higher priority command. This corresponds to a worst case decision as the sum of the two (non-scaled) commands could still be feasible even if the infinity norm of the higher priority one exceeds its threshold (refer to [12] for details).

The solution to overcome these limitations is described in the following. The novel approach will be described with reference to a multi robot system. In particular, for the sake of generality, first the novel saturation management technique will be applied to the velocity commands of each robot actuator, and then it will be specified for the application of the NSB approach to the motion control of mobile robots.

A. Novel Actuator Velocity Saturation Management Technique

1) Notation: There are l robots, each with n_i actuators for $i = 1, 2, \dots, l$. The total number of actuators is $n = \sum_{i=1}^{l} n_i$. Any regulation task h (that is to bring $\boldsymbol{\sigma}_h$ to its constant desired value $\boldsymbol{\sigma}_{h,d}$), that involves all the l robots (global task), would have dynamics

$$\dot{\boldsymbol{\sigma}}_h = \mathbf{J}_h(\mathbf{q}_h)\dot{\mathbf{q}}_h \tag{5}$$

where $\boldsymbol{\sigma}_h \in \mathbb{R}^{m_h}$, $\mathbf{q}_h \in \mathbb{R}^n$ and $\mathbf{J}_h \in \mathbb{R}^{m_h \times n}$. Notice that the Jacobian in eq. (5) is computed with respect to the system joint speeds $\dot{\mathbf{q}}_h$ as opposed to the one in eq. (1) that referred to the system's velocity $\mathbf{v}$. The ideal joint velocity command associated with this task is:

$$\dot{\mathbf{q}}_h = \mathbf{J}_h^{\dagger}\left(\boldsymbol{\Lambda}_h(\boldsymbol{\sigma}_{h,d} - \boldsymbol{\sigma}_h)\right). \tag{6}$$

If a task, the k^{th}, was to involve only a subset of robots (local task), its dynamics would have the same structure, but with a Jacobian of different sizes, for example

$$\dot{\boldsymbol{\sigma}}_k = \mathbf{J}_k(\mathbf{q}_k)\dot{\mathbf{q}}_k \tag{7}$$

with $\boldsymbol{\sigma}_k \in \mathbb{R}^{m_k}$, $\mathbf{q}_k \in \mathbb{R}^{\mu}$ and $\mathbf{J}_k \in \mathbb{R}^{m_k \times \mu}$. The velocity command $\dot{\mathbf{q}}_k \in \mathbb{R}^{\mu}$ associated with this task would be:

$$\dot{\mathbf{q}}_k = \mathbf{J}_k^{\dagger}\left(\boldsymbol{\Lambda}_k(\boldsymbol{\sigma}_{k,d} - \boldsymbol{\sigma}_k)\right). \tag{8}$$

In order to combine the commands of global and local tasks, it is necessary to construct the command vectors associated with local tasks with the same dimensions of the command vectors associated with the global tasks. These vectors will be indicated with an asterisk *, which in this case would be $\dot{\mathbf{q}}_k^* \in \mathbb{R}^n$. This vector is obtained putting in a column the components of $\dot{\mathbf{q}}_k$ into the positions related to the

corresponding actuators of the global task and assigning the value zero to all other components. In practice, for example:

$$\dot{\mathbf{q}}_k^* = (0, 0, \ldots, \dot{q}_{k1}, \dot{q}_{k2}, \ldots, 0, \ldots, \dot{q}_{k(\mu-1)}, \dot{q}_{k\mu}, 0, \ldots)^T$$

where $\dot{q}_{kj}$ is the j^{th} component of the vector $\dot{\mathbf{q}}_k$ (k^{th} task) and the vector $\mathbf{q}_k^* \in \mathbb{R}^n$ is ordered, meaning that its p^{th} component is always related to the same actuator of the same robot. In the absence of saturation, the classical NSB solution corresponding to N_{task} tasks (ordered with decreasing priority) would produce an overall command $\dot{\mathbf{q}}_{tot} \in \mathbb{R}^n$:

$$\dot{\mathbf{q}}_{tot} = \sum_{k=1}^{N_{task}} \dot{\mathbf{q}}_k^*. \tag{9}$$

2) Saturation Management: The solution in the eq. (9) does not account for the fact that the single actuators may be constrained to produce outputs in a limited range. In this respect, note that, by construction, if a command $\dot{\mathbf{q}}_k^*$ ensures that the task error $(\boldsymbol{\sigma}_{k,d} - \boldsymbol{\sigma}_k)$ goes to zero, the same is valid for any scaled command $\alpha_k \, \dot{\mathbf{q}}_k^*$ with $\alpha_k \in (0, 1]$[1].

With reference to the task 1 (task of highest priority), suppose that the j^{th} actuator should (or could) produce outputs only in the interval $[\, \underline{\dot{q}}_j, \overline{\dot{q}}_j \,]$: then the set of admissible task 1 commands can be introduced as

$$B_1 \;:=\; \left\{ \dot{\mathbf{q}}_1^* \in \mathbb{R}^n : \dot{q}_{1j}^* \in [\, \underline{\dot{q}}_j, \overline{\dot{q}}_j \,] \; \forall \; j = 1, \ldots, n \right\},$$

and the set of scaling-admissible commands is defined as:

$$\begin{aligned} S_1 \;:=\; \Big\{ & \dot{\mathbf{q}}_1^* \in \mathbb{R}^n : \exists \; \alpha_1 \in (0, 1] \Longrightarrow \\ & \alpha_1 \, \dot{q}_{1j}^* \in [\, \underline{\dot{q}}_j, \overline{\dot{q}}_j \,] \; \forall \; j = 1, \ldots, n \Big\}. \end{aligned}$$

If the set S_1 is empty, it means that the highest priority task is not compatible with the constraints on actuators. Therefore, it can not be implemented even in a scaled form. In particular, when the set S_1 is empty, given that task 1 cannot be realized, its corresponding command will need to be the null command, namely it can be written as $\overline{\alpha}_1 \, \dot{\mathbf{q}}_1^*$ with $\overline{\alpha}_1 = 0$. If the set S_1 is not empty, the vector $\overline{\alpha}_1 \, \dot{\mathbf{q}}_1^*$ can be commanded being $\overline{\alpha}_1 = \max\{\alpha \in (0, 1] : \alpha \, \dot{\mathbf{q}}_1^* \in B_1\}$.

Once the vector $\overline{\alpha}_1 \, \dot{\mathbf{q}}_1^*$ has been commanded, a similar procedure can be adopted for the task of priority $k \geq 2$, i.e.:

$$\begin{aligned} B_k \;&:=\; \left\{ \dot{\mathbf{q}}_k^* \in \mathbb{R}^n : \dot{q}_{kj}^* \in [\, \underline{c}_{kj}, \overline{c}_{kj} \,] \; \forall \; j = 1, \ldots, n \right\} \\ S_k \;&:=\; \Big\{ \dot{\mathbf{q}}_k^* \in \mathbb{R}^n : \exists \; \alpha_k \in (0, 1] \Longrightarrow \\ & \qquad \alpha_k \, \dot{q}_{kj}^* \in [\, \underline{c}_{kj}, \overline{c}_{kj} \,] \; \forall \; j = 1, \ldots, n \Big\} \end{aligned}$$

where

$$\begin{aligned} \underline{c}_{kj} \;&:=\; \underline{c}_{(k-1)j} - \overline{\alpha}_{(k-1)} \, \dot{q}_{(k-1)j}^* \\ \overline{c}_{kj} \;&:=\; \overline{c}_{(k-1)j} - \overline{\alpha}_{(k-1)} \, \dot{q}_{(k-1)j}^*. \end{aligned}$$

with $\underline{c}_{1j} := \underline{\dot{q}}_j$ and $\overline{c}_{1j} := \overline{\dot{q}}_j$. If the set S_k is empty, the task k is not commanded as it is incompatible with the constraints on actuators. In particular, in this case the null command $\overline{\alpha}_k \, \dot{\mathbf{q}}_k^*$, with $\overline{\alpha}_k = 0$, can be associated with the task k. If the set S_k is not empty, the command vector $\overline{\alpha}_k \, \dot{\mathbf{q}}_k^*$, with $\overline{\alpha}_k = \max\{\alpha \in (0, 1] : \alpha \, \dot{\mathbf{q}}_k^* \in B_k\}$, can be associated with the task k.

In the presence of saturation, the final command results

$$\dot{\mathbf{q}}_{tot} = \sum_{k=1}^{N_{task}} \overline{\alpha}_k \dot{\mathbf{q}}_k^* \tag{10}$$

where the scale factors $\overline{\alpha}_k$ are null or in $(0, 1]$, depending on whether the corresponding sets S_k are empty or not.

Note that by construction, eq. (10) guarantees that all command saturation constraints are always respected. Moreover, to the contrary of what occurs by implementing the solution in [8] and [12], it may happen that lower priority tasks are executed while some of their higher priority ones are not because unfeasible with respect to the actuator limits. In practice, this may occur because within the new proposed solution it may happen that $\overline{\alpha}_i = 0$ and $\overline{\alpha}_j > 0$ with $j > i$.

3) Saturation of Commands in Operational Space: The described procedure refers to the saturation management of actuators, that is (according to the typical notation of the industrial robotics) in the space of joints. This approach appears to be quite natural as, whatever should be the control architecture adopted, velocity saturations affect actuators. Nevertheless, velocity saturations issues may occur in operational space: in kinematics based guidance control loops of vehicles, for example, the produced guidance commands are vehicle speeds rather than actuator speeds.

Suppose that the l robots of interest are modeled as material points in the plane with single robot tasks formulated as in the eq. (1). In this scenario, it may be necessary to ensure that the commands $\mathbf{v}_{i,d} = (\dot{x}_{i,d}, \, \dot{y}_{i,d})^T$ given by eq. (3) for the i^{th} robot are bounded in Euclidean norm. Therefore, the Euclidean norm of the maximum possible velocity of the i^{th} robot is indicated with $v_{i\max}$. To ensure that the velocities of each material point are bounded in Euclidean norm by $v_{i\max}$, the procedure described in previous sections is implemented as follows:

- Calculate the classical NSB commands for all robots and all tasks as in the absence of saturation.
- Apply the procedure described above building all the N_{task} vectors $\dot{\mathbf{q}}_k^*$ as follows:

  $$\dot{\mathbf{q}}_k^* = (\bar{\dot{x}}_{1k}, \bar{\dot{y}}_{1k}, \bar{\dot{x}}_{2k}, \bar{\dot{y}}_{2k}, \ldots, \bar{\dot{x}}_{lk}, \bar{\dot{y}}_{lk})^T$$

  where $(\bar{\dot{x}}_{ik}, \, \bar{\dot{y}}_{ik})^T = \bar{\mathbf{v}}_{ik}$ is defined as in the eq. (4). Having assumed the presence of l robots, the overall command vector will have $n = 2l$ components. The maximum and minimum thresholds of saturation related to the task of priority 1 are defined as:

  $$\begin{aligned} \overline{\dot{q}}_{2i-1} = v_{i\max} \quad \overline{\dot{q}}_{2i} = v_{i\max} \quad \forall \; i = 1, \ldots, l \\ \underline{\dot{q}}_{2i-1} = -v_{i\max} \quad \underline{\dot{q}}_{2i} = -v_{i\max} \quad \forall \; i = 1, \ldots, l \end{aligned}$$

  where $v_{i\max}$ and $-v_{i\max}$ are respectively the maximum and minimum thresholds for both $\dot{x}_{i,d}$ and $\dot{y}_{i,d}$.
- Apply the procedure described in the previous sections to obtain a final command $\dot{\mathbf{q}}_{tot}$ given by eq. (10).

[1] true also for $\alpha_k > 1$, but α is limited to the range $(0, 1]$ to avoid amplifying the control command w.r.t. the choice of the gains $\boldsymbol{\Lambda}_k$

By construction, each couple $(\dot{q}_{(2i-1),tot}, \dot{q}_{2i,tot})^T$ extracted from the vector $\dot{\mathbf{q}}_{tot}$ corresponds to a velocity commands $\mathbf{v}_{i,d}$ for a single robot $i = 1, 2, \ldots, l$. To ensure that the Euclidean norms of the velocity commands of each robot are bounded by the corresponding $v_{i\max}$, it may still be necessary to scale the command $\dot{\mathbf{q}}_{tot}$ by a proper factor $\beta \in (1/\sqrt{2}, 1)$. A conservative solution that minimizes the computational burden could be simply to divide the command $\dot{\mathbf{q}}_{tot}$ given by eq. (10) by $\sqrt{2}$. Finally, the overall velocity command will be given by:

$$\dot{\mathbf{q}}_{tot} = \beta \left(\sum_{k=1}^{N_{task}} \overline{\alpha}_k \dot{\mathbf{q}}_k^* \right). \tag{11}$$

IV. Case study

In order to test the conjuncted action of the NSB with the saturation management technique described in section III, this section analyzes the case study of a team of six mobile robots that has to entrap and to escort a target placing the team members at the vertices of a regular polygon centered in the target.

A. The Escorting Mission

The escorting mission, described in detail in [2], is decomposed in four elementary tasks:

- Task 1: *Obstacle avoidance*. Each vehicle needs to avoid both environmental obstacles and the other vehicles. With reference to the generic vehicle of the team, in presence of a punctual obstacle in the advancing direction, the task aim is to keep the robot at a safe distance d from the obstacle. Therefore, defining as $\boldsymbol{p}_o$ the obstacle position, it is:

$$\sigma_o = \|\boldsymbol{p} - \boldsymbol{p}_o\| \in \mathbb{R}; \quad \sigma_{o,d} = d.$$

- Task 2: *Distribution on a circumference*. This task moves the robots on a given circumference around the centroid. The relative l-dimensional task function is:

$$\boldsymbol{\sigma}_s = \begin{bmatrix} \ldots & \frac{1}{2}(\boldsymbol{p}_i - \boldsymbol{c})^T(\boldsymbol{p}_i - \boldsymbol{c}) & \ldots \end{bmatrix}^T$$

used to keep each robot of the team at a given distance r from a point $\boldsymbol{c} \in \mathbb{R}^2$ by setting

$$\boldsymbol{\sigma}_{s,d} = \begin{bmatrix} \ldots & \frac{r^2}{2} & \ldots \end{bmatrix}^T.$$

- Task 3: *Centroid on the target*. This task commands the robots' centroid to be coincident with the target. The two-dimensional task function $\boldsymbol{\sigma}_c$ is simply given by:

$$\boldsymbol{\sigma}_c = \mathbf{f}_c(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_l) = \frac{1}{l}\sum_{i=1}^{l} \boldsymbol{p}_i = \bar{\boldsymbol{p}}.$$

- Task 4: *Polygon with equal edges*. The polygon-with-equal-edges task distributes the robot at the vertices of a polygon with equal sides. This is achieved by simply imposing the same distance between adjacent vehicles.

B. Unicycle-like Kinematic Model

Assume that all vehicles to be controlled are differential drive wheeled mobile robots moving in a horizontal plane. In order to implement on such mobile robots the kinematics control solution described in section II, the systems' velocity $\mathbf{v}_d$ should be assigned at each time instant. Since the NSB outputs a linear velocity for a material point, while the mobile robot actuators are its wheels, it is necessary to convert the NSB output $\mathbf{v}_d$ to wheels' desired velocities $\boldsymbol{\omega}$.

As described in detail in [8], given the non-holonomic nature of the mobile robot, the velocity of the wheels axis center C cannot be arbitrarily assigned. To overcome this difficulty, the proposed control solution can be applied, as in [13], to a point P that lies a distance Δ_P along the normal to the wheel axis and intersect the wheel axis in the point C. Denoting with $\mathbf{v}_P = (v_P(x), v_P(y))^T$ the velocity of point P in the absolute frame, standard kinematics relations allow to express wheels' speeds $(\omega_l, \omega_r)^T$ in terms of $\mathbf{v}_P$ as:

$$\begin{pmatrix} \omega_l \\ \omega_r \end{pmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -\frac{b}{\Delta_P} \\ 1 & \frac{b}{\Delta_P} \end{bmatrix} \begin{bmatrix} c_\theta & s_\theta \\ -s_\theta & c_\theta \end{bmatrix} \begin{pmatrix} v_P(x) \\ v_P(y) \end{pmatrix} \tag{12}$$

being ω_l and ω_r the left and right wheels angular velocity respectively, r the radius of the wheels, $2b$ the distance between wheels, $c_\theta := \cos\theta$ and $s_\theta := \sin\theta$.

C. Simulation results

The simulations have been performed via an ad-hoc software simulator written in C language that uses the same parameters that characterize the experimental mission. The chosen parameters are summarized in the following table:

Kinematic parameters	Mission parameters
r = $8mm$	$v_{i\max}$ = $15cm/s$
b = $2.65cm$	$\sigma_{o,d}$ = $20cm$
$\frac{\Delta_p}{b} = 1$	$\Lambda_o = 1$, $\Lambda_s = 0.5$
	$\Lambda_c = 2$, $\Lambda_p = 3$

In the proposed mission, a team of six robots has to entrap and to escort a target that is a yellow disc. Fig. 1 shows the system evolution during the whole simulation. In particular, at the beginning of simulation, the robot n. 6 is deeply inside the safety region (dotted line) of the disc placed in the point of coordinates (80, 110), while the others five robots are aligned parallel to the x-axis. After 5.65 seconds the six robots have entrapped the target. Then, the disc is moved in a straight line to the point of coordinates (76, 70). Therefore, the team follows and escorts the disc until destination.

To describe how the proposed approach operates, the attention is focused on the behavior of the robot n. 4. Fig. 2 shows the norm of the velocity commands for each task of the robot in the absence (dotted line) and in the presence (solid line) of saturation. In particular, Fig. 2 show the norm of $\bar{\mathbf{v}}_{4k}$ and $\overline{\alpha}_k\bar{\mathbf{v}}_{4k}$ for $k = 1, \ldots, 4$. Fig. 3 shows the norm of the overall velocity command $\mathbf{v}_{4,d}$ during the simulation. It is worth noticing that the Euclidean norm of $\mathbf{v}_{4,d}$ is always under the maximum threshold $v_{4\max} = 15cm/s$. Finally, Fig. 4 shows the wheels' angular velocities ω_l (in blue) and ω_r (in red) of the robot during the overall simulation.

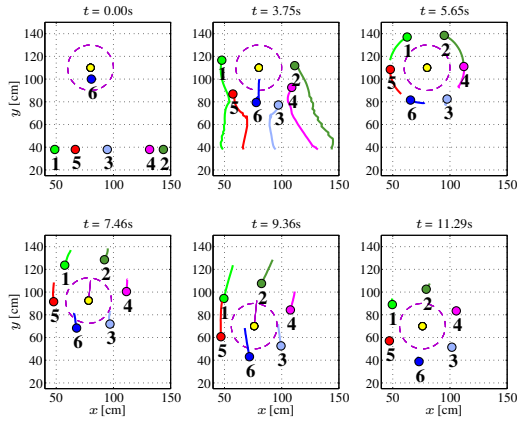

Fig. 1. Snapshots of the system evolution during the simulation.

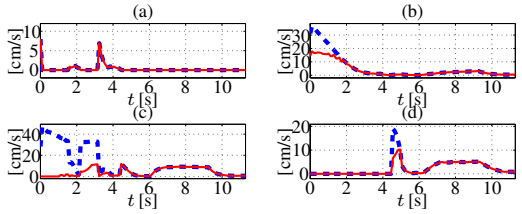

Fig. 2. Task velocity commands for the robot n. 4 during the simulation. Norm of $\bar{\mathbf{v}}_{4k}$ in dotted line and the norm of $\overline{\alpha}_k\bar{\mathbf{v}}_{4k}$ in solid line for the tasks: a) obstacle-avoidance; b) distribution-on-a-circumference; c) centroid-on-the-target; d) polygon-with-equal-edges.

D. Experimental results

The experimental platform bases on a team of Khepera II mobile robot, showed in Fig. 5, manufactured by K-team [1] and available at the LAI (Laboratorio di Automazione Industriale) of the Università degli Studi di Cassino. Each robot is a differential-drive mobile robots with an approximate dimension of 8 cm of diameter. Each robot can communicate trough a Bluetooth module with a remote Linux-based PC where the NSB has been implemented. To allow the needed absolute position measurements we have developed a vision-based system using two CCD cameras. The remote PC receives the position measurements at a sampling time of 70 ms and elaborates the NSB control. Once the NSB outputs the desired linear velocities, the wheels' desired velocities are elaborated referring to the kinematic model of Section IV-B and sent to the robot through the Bluetooth module.

Analogously to the simulative case study, Fig. 6 shows the system evolution during the whole simulation. Moreover, a video of the experiment accompanies this paper. The behaviors of the numerical simulations and of the experiments are in good agreement. To properly evaluate the saturation mechanism technique, we will focus the attention on the behavior of the robot n. 6. Fig. 7 shows the main saturation parameters for all tasks of this robot during the experiment. In particular, Fig. 7.a-b-c-d show the norm of $\bar{\mathbf{v}}_{6k}$ to the left, $\overline{\alpha}_k$ in the middle and the norm of $\overline{\alpha}_k\bar{\mathbf{v}}_{6k}$ to the right respectively for $k = 1, \ldots, 4$. Fig. 8 shows the norm of

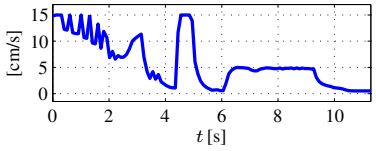

Fig. 3. Norm of the velocity $\mathbf{v}_d$ of the robot n. 4 during the simulation.

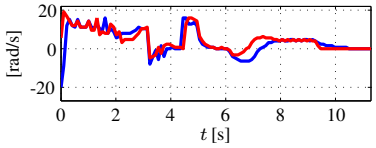

Fig. 4. Wheels' angular velocities of the robot n. 4 during the simulation.

the overall velocity command $\mathbf{v}_{6,d}$ for the robot during experiment. Finally, Fig. 9 shows the task errors for the whole system during the experiment. The errors are first convergent to zero. Then, when the target is moved, the errors increase and finally converge again at zero since the robots follow and escort the target to destination.

V. Conclusions

In this paper, the conjunct application of a behavior-based technique, namely the Null-Space based Behavioral control [3], with the new saturation management technique described in section III, has been investigated. The scope of this paper was to extend the Null-Space based Behavioral control to the motion control of robotic systems with velocity saturated actuators, avoiding that velocity saturations induced by lower priority tasks corrupt the higher priority ones. In particular, the proposed solution aims at managing actuator velocity saturations by dynamically scaling task velocity commands, so that the hierarchy of task priorities is preserved in spite of actuator velocity constraints, and at overcoming the limits of the solution proposed in [12]. The proposed approach has been tested on the motion control of a team of cooperative mobile robots, and it has been validated both by numerical simulations and experimental results.

VI. Acknowledgements

The research leading to these results has received funding from the European Communitys Seventh Framework Programme project CHAT - "Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security", Contract Number INFSO-ICT-224428.

References

[1] http://www.k-team.com/. K-Team.

[2] G. Antonelli, F. Arrichiello, and S. Chiaverini. The entrapment/escorting mission: An experimental study using a multirobot system. *IEEE Robotics and Automation Magazine (RAM). Special Issues on Design, Control, and Applications of Real-World Multi-Robot Systems*, 15(1):22–29, March 2008.

[3] G. Antonelli, F. Arrichiello, and S. Chiaverini. The Null-Space-based Behavioral control for autonomous robotic systems. *Journal of Intelligent Service Robotics*, 1(1):27–39, online March 2007, printed January 2008.

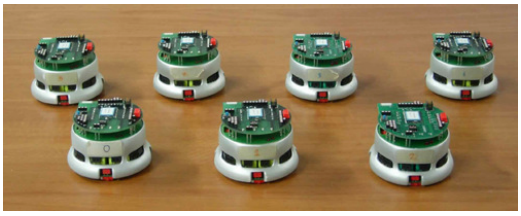

Fig. 5. Team of Khepera II mobile robots used during the experiment.

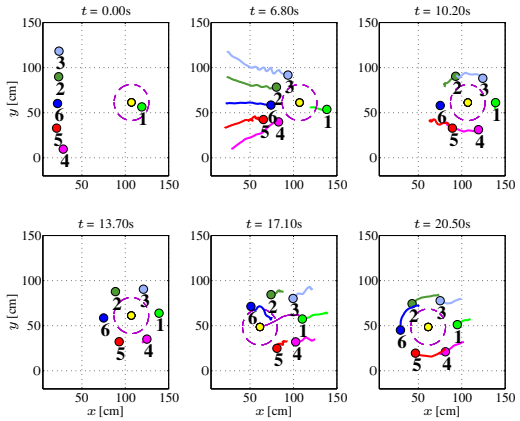

Fig. 6. Snapshots of the system evolution during an experiment performed with 6 Khepera II robots.

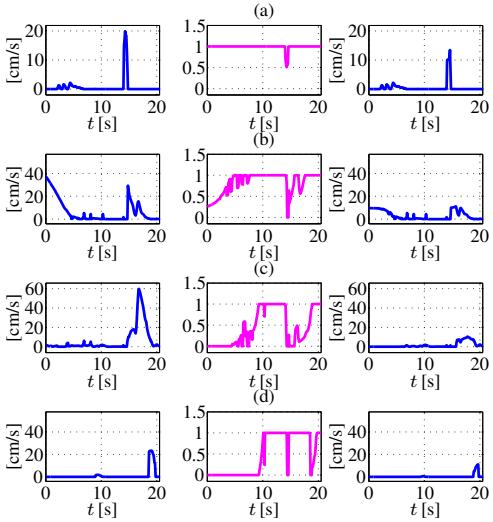

Fig. 7. Parameters of the saturation management technique for the robot n. 6 during the whole experiment. The rows are relative to the tasks function: a) obstacle-avoidance; b) distribution-on-a-circumference; c) centroid-on-the-target; d) polygon-with-equal-edges tasks. In all the cases, the left, middle and right plots respectively show the norm of the velocity command $\bar{\mathbf{v}}_k$, the scaling factor $\overline{\alpha}_k$ and the norm of the scaled velocity command $\overline{\alpha}_k\,\bar{\mathbf{v}}_k$.

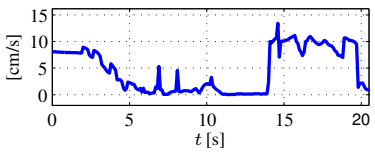

Fig. 8. Norm of the velocity $\mathbf{v}_d$ of the robot n. 6 during the experiment.

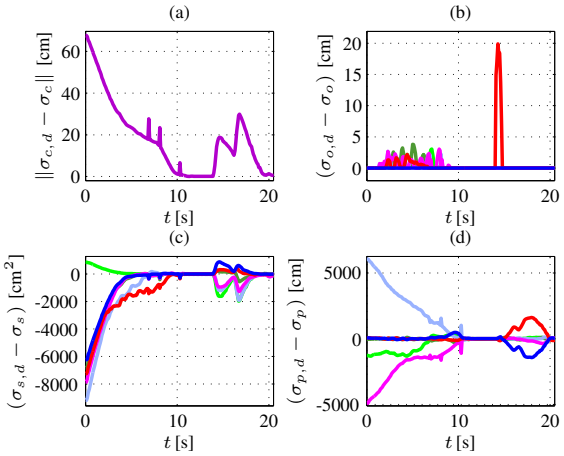

Fig. 9. Errors of the task functions during the experiment. a) Centroid; b) Obstacle-avoidance; c) Distribution-on-a-circumference; d) Polygon-with-equal-edges.

[4] G. Antonelli, F. Arrichiello, and S. Chiaverini. Stability analysis for the null-space-based behavioral control for multi-robot systems. In *47th IEEE Conference on Decision and Control and 8th European Control Conference*, Cancun, Mexico, Dec. 2008.

[5] G. Antonelli and S. Chiaverini. Kinematic control of a platoon of autonomous vehicles. In *Proceedings 2003 IEEE International Conference on Robotics and Automation*, pages 1464–1469, Taipei, TW, Sept. 2003.

[6] R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.

[7] R.C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, MA, 1998.

[8] F. Arrichiello, S. Chiaverini, P. Pedone, A. A. Zizzari, and G. Indiveri. The null-space based behavioral control for non-holonomic mobile robots with actuators velocity saturation. In *The 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17 2009.

[9] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[10] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[11] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.

[12] G. Indiveri. Swedish wheeled omnidirectional mobile robots: kinematics analysis and control. *IEEE Transactions on Robotics*, 25:164–171, Feb. 2009.

[13] JRT Lawton, RW Beard, and BJ Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, 2003.

[14] A.A. Maciejewski. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, 5(6):527–552, 1988.

[15] M.J. Matarič. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.

[16] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal Robotics Research*, 6(2):3–15, 1987.

[17] L.E. Parker. On the design of behavior-based multi-robot teams. *Advanced Robotics*, 10(6):547–578, 1996.

[18] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: The Task Function Approach*. Clarendon Press, oxford engineering science series no. 22 edition, 1991.