

TS Fuzzy Null-Space Behavior Control of Non-Holonomic Robot Formations

Gustavo QUINTANA-CARAPIA^{1,*}, Jorge S. BENÍTEZ-READ²,
J. Armando SEGOVIA-DE-LOS-RÍOS³, Mayra P. GARDUÑO-GAFFARE⁴

¹⁻⁴ Instituto Tecnológico de Toluca (ITT),
Av. Tecnológico s/n. La Virgen, Metepec, 52149, México.

¹⁻³ Instituto Nacional de Investigaciones Nucleares,
Carretera México-Toluca s/n,
La Marquesa, Ocoyoacac, 52750, México
gustavo.quintana@inin.gob.mx; jorge.benitez@inin.gob.mx; armando.segovia@inin.gob.mx

* corresponding author

Abstract: In this paper, an improved Null Space behavior control of a team of non-holonomic robots that maintains time-varying formations is presented. The task functions: (a) are combined considering the non-holonomic constraints, (b) are designed such that each robot tracks its desired trajectory by synchronizing its motion with the other robots motion, (c) maintain the kinematics relationships required by the formation, (d) include synchronization constraints and synchronization errors, which are a measurement of the formation realization degree that is used to generate suitable actions to reach the goal formation, and (e) are in charge of orientating the robot towards its desired position during shape switching. In order to ensure the operation of the controller, a Takagi-Sugeno (TS) fuzzy system is implemented to keep the task functions solutions below the saturation values determined by the actuators of the differential mobile robots. Finally, simulations and experiments are performed to demonstrate the effectiveness of the proposed Null Space behavior control for non-holonomic robots approach in formation control tasks.

Keywords: formation control; inverse kinematics; null space based behaviour control.

1. Introduction

Nowadays, the field of study of autonomous mobile robotics is very fertile. The topic is attractive due to the fact that certain jobs can be performed faster and better by a group of robots working as a team [1, 2]. A seminal work, where virtual agents controlled their formation by following simple rules [3], inspired the robot formation control. In a robot formation, the robots in the group are able to maintain predefined positions among them while the group moves as if it were a single individual [4]. Research in robot formation aims to provide control to several robots [5], joining the different tasks that each robot can assume, depending on its current configuration.

Formation control can be achieved with behavior control. The main idea in behavior control is to simulate the biological and social interactions that occur in animal species with artificial beings [6]. To obtain this, the general problem is decomposed into several sub-problems (denominated behaviours) that are solved simultaneously. The solutions of the sub-problems are then used to assemble the next robot motion commands.

The main difficulty in behaviour control is the asynchronous processing termination of each sub-problem, which can lead to wrong

command orders. In order to reduce the uncertainty caused by the asynchronous behaviour termination, special attention must be paid to the composition of the results.

One approach, the competitive method introduced in [7], considers only one behaviour for the generation of the control command. In this case, the different behaviours compete to be the one and only that determines the command. A different approach was proposed in [8] to consider the contribution of the entire set of behaviours, by means of a weighted sum, to obtain the control command.

The null space based method was introduced in the seminal work of [9] and combines both the competitive and cooperative paradigms. In the null space based method, the behaviours are described using kinematics task functions, which are prioritized in terms of their relevance for the objectives of the application, as in the competitive method [10]. The results of the different task functions are combined, cooperatively, by projecting each behaviour into the null space of the following task function in the order of the ascending hierarchy.

In this paper, an improved null-space based behaviour (NSB) approach is proposed to address the formation control of non-holonomic robots. The new features incorporated in the control scheme are described as follows: First,

a successful modification of the NSB control to include the orientation as an additional argument in the definition of the task functions. This feature allows the use of the robot orientations as an important factor in the formation control. Second, the development of a decentralized NSB controller for non-holonomic robots, that considers the relative movement of the other members in the formation. The control algorithm requires local information of two neighbouring robots. Third, simulations are performed in groups of mobile robots to show the effectiveness of the proposed formation control using a generalized super-ellipse, whose parameters are functions of time and gives a variety of reference shapes.

2. NSB Behavior Control for Non-Holonomic Robots

The kinematics of a non-holonomic differential mobile robot, with two driving standard wheels, is given as [11]:

$$\begin{aligned}\dot{x} &= \frac{1}{2}(v_L + v_R) \cos \theta = v \cos \theta \\ \dot{y} &= \frac{1}{2}(v_L + v_R) \sin \theta = v \sin \theta \\ \dot{\theta} &= \frac{1}{2l}(v_L - v_R) = \omega\end{aligned}\quad (1)$$

where $\mathbf{q} = [x, y, \theta]^T$ is the configuration of the robot that contains the position of the wheels axis centre (x, y) and the robot body orientation θ in the global reference frame. The length l is the distance from the point (x, y) to each of the wheels' rotation centre. The velocities v_L and v_R are the tangent velocities of the left and right wheels, respectively, whereas v and ω are the tangent and angular velocities of the mobile robot. In matrix form, the robot velocities are related in the following bearing [11]:

$$\begin{bmatrix} v_L \\ v_R \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & l \\ \cos \theta & \sin \theta & -l \\ -\sin \theta & \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (2)$$

It can be observed that in order to control the motion of the mobile robot in its workspace, suitable velocities have to be applied to both left and right wheels. To obtain the wheel velocities, in this paper we use the procedure described as follows.

The null space based behaviour control expresses the different behaviours by making use of so-called task functions. Each task

function $\sigma = \mathbf{f}(\mathbf{q})$ takes its argument from the current robot configuration. The first derivative of a task function leads to a direct relationship between the task rate of change $\dot{\sigma}$ and the robot velocities (1) where the Jacobian matrix $\mathbf{J}(\mathbf{q})$ is the proportionality constant:

$$\dot{\sigma} = \frac{\partial f(q)}{\partial q} = J(q)\dot{q} \quad (3)$$

From expression (3) the velocities \dot{q} can be obtained in terms of the task rate of change $\dot{\sigma}$. It is necessary to invert the Jacobian matrix $\mathbf{J}(\mathbf{q})$ [9], or the pseudoinverse $\mathbf{J}^\dagger(\mathbf{q})$ in case the Jacobian matrix is not square. In this latter case, the least squares method matrix provides a unique solution [12]:

$$\dot{q} = J^\dagger \dot{\sigma} = J^T (\mathbf{J} \mathbf{J}^T)^{-1} \dot{\sigma} \quad (4)$$

Therefore, to obtain desired velocities for the robot, the task function rate of change shall be defined as a suitable value. Another factor that is important to consider in the definition of the desired velocities \dot{q} is the instantaneous value of the task function. If the task function current value σ is compared to a certain preset value σ_d , the error $\tilde{\sigma} = \sigma_d - \sigma$ is a valuable resource for the design of the task function. The task function can be designed in such a way that during the robots displacement, the instantaneous value of σ changes towards the value of σ_d , the error $\tilde{\sigma}$ decreases and the behavior is being achieved:

$$\dot{q} = J^\dagger (\dot{\sigma}_d + \Lambda \tilde{\sigma}) \quad (5)$$

where Λ is a diagonal positive matrix of gains. If the expression shown in (5) is included in (2), then this method can be used in the motion control of a non-holonomic mobile robot through the use of the following equation:

$$\dot{\phi} = G(q) J^\dagger(q) (\dot{\sigma}_d + \Lambda \tilde{\sigma}) \quad (6)$$

where $\dot{\phi} = [v_L, v_R, 0]^T$ and \mathbf{G} is the matrix shown in (2). The robot formation control problem can be designed as a combination of several task functions. The different behaviors contributions are considered to create the motion commands. In the NSB behavior control approach, the task functions are hierarchically sorted in terms of priority for the robot formation. The task with the highest priority is assigned to be the primary task. Following the sorting order, tasks of lower priorities are assigned to be secondary, tertiary and so on. The velocities of the i_{th} -priority task can be expressed as:

$$\dot{\phi}_i = \mathbf{GJ}_i^\dagger (\dot{\sigma}_{i,d} + A_i \tilde{\sigma}) \quad (7)$$

The hierarchical arrangement of the tasks functions is useful to combine the resulting behavior velocities. In the NSB behavior control approach, the behavior velocities are projected into the null space of the immediate higher priority task, as follows:

$$\dot{\phi}_d = \dot{\phi}_1 + N_1 (\dot{\phi}_2 + N_2 (\dot{\phi}_3 + \dots)) \quad (8)$$

where the i_{th} -priority task null space is defined as $N_i = I - \mathbf{GJ}_i^\dagger J_i G^{-1}$. The null space projection aggregates the contribution of each task when its velocities do not counteract the task functions of higher priority. If there are conflicts of contradiction between the velocities of two or more task functions, then the contributions of the lower priority tasks are not aggregated in the final value

3. Design of NSB Behaviors

The configuration of a group of two-wheeled mobile robots is shown in Figure 1. The configuration of the i_{th} robot in the global reference frame is represented by $\mathbf{q}_i = [x_i, y_i, \theta_i]^T$ where (x_i, y_i) are the position coordinates and θ_i is the orientation angle. The formation control problem is considered as the guidance and positioning of a group of n robots along the boundary of a planar curve. The desired time-varying shape for the robot formation is denoted by $\mathbf{S}(\mathbf{p}, t)$, where \mathbf{p} contains the coordinates of the points on the boundary of \mathbf{S} in time t . The boundary of \mathbf{S} is parameterized by the planar curve $\partial\mathbf{S}(\mathbf{p}, t) = 0$ and the i_{th} robot target position, $\mathbf{p}_{i,d}$, is assigned to be on the boundary shape such that $\partial\mathbf{S}(\mathbf{p}, t) = 0$. In this section is presented the design of suitable task functions such that each robot in the group converges to its goal position and simultaneously the formation acquires the desired shape. Even though the definition of the boundary does not imply a certain orientation for the robots, it is recommended for the task function to include a mean to explicitly regulate each robot orientation.

In this work, the modified null space based behavior control is used to solve the formation control problem. New task functions are introduced to govern the movements as they track the target positions that satisfy $\partial\mathbf{S}(\mathbf{p}_i, t) = 0$. The task functions are designed for decentralized topologies.

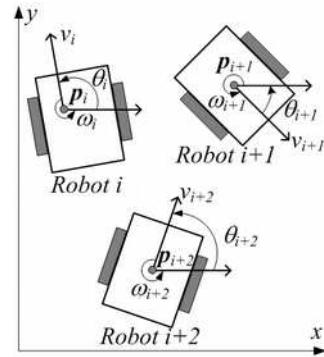


Figure 1. Robot configuration parameters.

In the decentralized mode of operation, it is required that the robots adopt the formation autonomously. Thus, it is needed that each robot runs its own independent control algorithm. The controller on board takes its decisions based on local information that is obtained from sensing and from information interchanged with the other robots in the team.

The NSB behaviour control algorithm is presented in Figure 2. The task functions, in the decentralized approach, represent behaviors for single robots that make only use of local information relative to the current robot and its neighbors. The formation is a result of the interactions of individual robots with the other members the team that are running the same control algorithm on board.

```
While (FormationNotYetAchieved) Do:
    Read local configuration q_i
    Read neighbors configurations q_Ni
    Evaluate task functions σ_i and Jacobian matrices J_i
    Adapt, via fuzzy system, gain matrices A_i
    Obtain the behavior velocities φ_i
    Project the behavior velocities into φ_d
    Apply wheel velocities φ_d to the robot
```

Figure 1. NSB decentralized controller algorithm.

Three task functions are proposed for the robots to adopt the formation along the boundary of a varying in time planar curve in decentralized mode.

The first task function, σ_1 , is expressed as follows:

$$\sigma_1 = \begin{bmatrix} 1 / \left(1 + e^{-k_{s1} (|p_o - p_R| - d_{tol})} \right) \\ 1 / \left(1 + e^{-k_{s2} (\arctan(p_o, p_R) - \alpha_{tol})} \right) \end{bmatrix} \quad (9)$$

Task function σ_1 has the highest priority and is in charge of supervising the distance from the closest neighbour robot O to the current robot R that is running the NSB controller. This task function avoids collisions between team-mates by keeping each robot separated by a distance

bigger than d_{TOL} . A review on the collision avoidance basics can be found in [13].

Task function σ_1 also changes the robot orientation in a repulsive way when there is a possible collision in front of the robot according to its current motion along its trajectory. The index of identity assigned to the robots is sorted according to the increasing counter-clockwise angles of the locations of the robots along the boundary of the shapes.

The value of the components of σ_1 is 1 when there is no danger of collision with the teammates and decreases towards 0 as they get closer. For this reason, the desired values of σ_1 are $\sigma_{1,d} = [1, 1]^T$.

The Jacobian matrix, \mathbf{J}_1 that results after taking the first derivative of σ_1 is the following:

$$\mathbf{J}_1 = \begin{bmatrix} \frac{J_{1n}(p_{Ox} - p_{Rx})}{-d_{OR}(J_{1d})^2} & \frac{J_{1n}(p_{Oy} - p_{Ry})}{-d_{OR}(J_{1d})^2} & 0 \\ \frac{J_{2n}(p_{Oy} - p_{Ry})}{(d_{OR}J_{2d})^2} & \frac{J_{2n}(p_{Ox} - p_{Rx})}{-(d_{OR}J_{2d})^2} & \frac{-J_{2n}}{(J_{2d})^2} \end{bmatrix} \quad (10)$$

considering that: $J_{1n} = k_{s1} e^{-k_{s1}(\|p_o - p_r\| - d_{tol})}$,

$$J_{2n} = k_{s2} e^{-k_{s2}(\arctan(p_o, p_r) - a_{tol})},$$

$$J_{1d} = 1 + e^{-k_{s1}(\|p_o - p_r\| - d_{tol})},$$

$$J_{2d} = 1 + e^{-k_{s2}(\arctan(p_o, p_r) - a_{tol})}, \text{ and}$$

$$d_{OR} = \|p_o - p_r\|.$$

The matrix \mathbf{J}_1^\dagger is obtained by the Moore-Penrose method. The relationship between the first task function and its associated velocities is obtained from (7), with $i = 1$.

To establish the second task function, σ_2 , we consider that the coordinates for the position and the angle of orientation for the i^{th} robot along the boundary of a generalized superellipse with varying parameters can be expressed as [14]:

$$\begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} (\cos \varphi_i(t))^{m(t)} & 0 & 0 \\ 0 & (\sin \varphi_i(t))^{m(t)} & 0 \\ 0 & 0 & \varphi_i(t) \end{bmatrix} \begin{bmatrix} a(t) \\ b(t) \\ 1 \end{bmatrix} \quad (11)$$

where a and b denote the longest and the shortest radii of the ellipse, respectively, and φ_i denotes the angle of the robot lying on the boundary with respect to the center of the ellipse. Equation (11) can also be expressed as $[x_i, y_i, \theta_i]^T = \mathbf{A}_i(t)[a(t), b(t), 1]^T$. Any of these parameters can be time-varying. The inverse of

matrix $\mathbf{A}_i(t)$ is a coupling parameter of the synchronization constraint to $\mathbf{q}_i = [x_i, y_i, \theta_i]^T$ [14]:

$$A_1^{-1} q_1 = A_2^{-1} q_2 = \dots = A_i^{-1} q_i = [a(t), b(t), 1]^T \quad (12)$$

The synchronization constraints are also applicable to the desired positions and orientations:

$$A_1^{-1} q_{1,d} = A_2^{-1} q_{2,d} = \dots = A_i^{-1} q_{i,d} = A_n^{-1} q_{n,d} \quad (13)$$

The difference of the synchronization constraints (13) and (12), $\mathbf{q}_{i,e} = \mathbf{q}_{i,d} - \mathbf{q}_i$, is a measure of the deviation of the robot configuration from the boundary of the ellipse:

$$A_1^{-1} q_{1,e} = A_2^{-1} q_{2,e} = \dots = A_i^{-1} q_{i,e} = A_n^{-1} q_{n,e} \quad (14)$$

The configuration synchronization error is denoted as ε_i and is defined as a combination of all available pairs of neighbour robots in the following manner

$$\begin{aligned} \varepsilon_1 &= A_1^{-1} q_{1,e} - A_2^{-1} q_{2,e} \\ \varepsilon_2 &= A_2^{-1} q_{2,e} - A_3^{-1} q_{3,e} \\ \dots & \\ \varepsilon_n &= A_n^{-1} q_{n,e} - A_1^{-1} q_{1,e} \end{aligned} \quad (15)$$

The formation control objective is achieved when $\varepsilon_i = \mathbf{0}$ for all the robots in the formation.

Function σ_2 is designed to measure the synchronization constraints errors existing between the current configuration of robot R and its desired configuration on the boundary of the shape:

$$\sigma_2 = \begin{bmatrix} A_R^{-1}(q_{i,d} - q_i) \\ \arctan(p_{i,d}, p_i) - \theta \end{bmatrix} \quad (16)$$

The value of this task function is a measurement of the formation effect. Given that the errors are required to converge to zero while the formation approaches the target shape, the desired values of σ_2 are $\sigma_{2,d} = \mathbf{0}$.

The Jacobian matrix \mathbf{J}_2 is expressed as follows:

$$J_2 = \begin{bmatrix} (\cos \varphi_i(t))^{-m(t)} & 0 & 0 \\ 0 & (\sin \varphi_i(t))^{-m(t)} & 0 \\ -(p_{i,dy} - p_{iy})/d_{ID} & -(p_{i,dx} - p_{ix})/d_{ID} & 1 \end{bmatrix} \quad (17)$$

where $d_{ID} = \|\mathbf{p}_{i,d} - \mathbf{p}_i\|$. The relationship between the second task function and its associated velocities is obtained from (7), with $i = 2$.

Function σ_3 measures the synchronization errors existing between the current robot R and its neighbour N. The value of this task function is also a measurement of the formation effect.

$$\sigma_3 = \varepsilon_i - \varepsilon_{i-1} \quad (18)$$

Given that the synchronization errors must converge to zero while the robot formation is approaching the desired shape along the boundary of the ellipse, the desired values of σ_3 are $\sigma_{3,d} = \mathbf{0}$.

The Jacobian matrix, \mathbf{J}_3 , corresponding to the first derivative of the third task function is:

$$J_3 = -2 \begin{bmatrix} (\cos \varphi_i(t))^{-m(t)} & 0 & 0 \\ 0 & (\sin \varphi_i(t))^{-m(t)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Matrix \mathbf{J}_3^\dagger is obtained by the Moore–Penrose method and the relationship between the third task function and its associated velocities is obtained from (7), with $i = 3$.

The global behaviour for each robot results from the combined velocities by projecting the lower priority velocities on the null space of the higher priority velocities as described in (8)

4. TS Fuzzy Adaptation of NSB Behaviours

One problem that can arise in the application of the NSB control is the saturation of the actuators as a consequence of an excess of velocity of one or several behaviour velocities. The velocity saturations of lower-priority behaviours negatively impact on the final control command as they corrupt the higher-priority behaviours [15]. In this paper, the actuator velocity saturations are avoided by a dynamic adaptation of the behaviours velocities, preserving the hierarchy and the NSB control scheme. The dynamic adaptation of the behaviours velocities is performed by Takagi-Sugeno (TS) fuzzy systems, which are an efficient soft computing technique to solve several problems in robotics [16].

The elements of the diagonal matrix Λ in equation (6) relate directly the task function instantaneous error $\tilde{\sigma}$ with its associated velocities $\dot{\varphi}$. In previous works [12, 14], the matrix Λ elements are kept constant during the robot transitions. In some cases, these constant gains may yield large velocities $\dot{\varphi}$. The combination of large velocities given by equation (8) can lead to unrealizable velocities if they exceed the saturation limit of the robot motors. One contribution of this work is to adapt the Λ_2 and Λ_3 elements through fuzzy systems. The aim of this adaptation process, which varies the values of the above mentioned

elements, is to prevent the saturation of the robots actuators. The TS fuzzy system, shown in Figure 3, maintains the velocities of the robots motors within the working intervals.

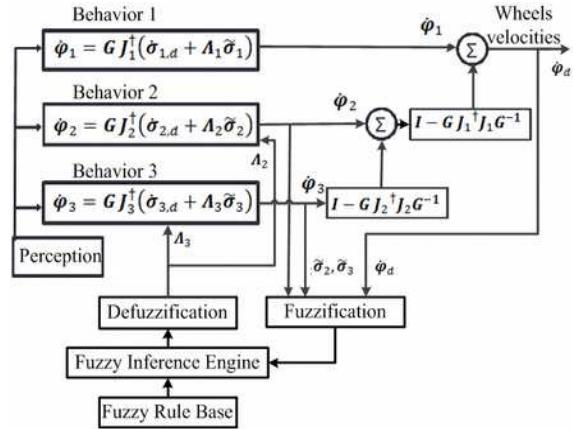


Figure 3. Behaviours adapted by fuzzy logic.

4.1 Adaptation of matrix Λ_2 elements

Matrix Λ_2 elements are adapted by the TS fuzzy system (see figure 3), which has five inputs and three outputs. The inputs are the components of the task function error vector $\tilde{\sigma}_2$, i.e. $\tilde{\sigma}_{2x}$, $\tilde{\sigma}_{2y}$ and $\tilde{\sigma}_{2\theta}$, and the left and right robot wheel velocities, v_L and v_R . The outputs of the TS fuzzy system are the Λ_2 elements, λ_{2x} , λ_{2y} and $\lambda_{2\theta}$. In general terms, the TS fuzzy system regulates the velocities $\dot{\varphi}_2$ reducing the Λ_2 elements when the error $\tilde{\sigma}_2$ is large or when the current v_L and v_R are large and close to saturation. Otherwise, the Λ_2 elements are increased to support the convergence of error $\tilde{\sigma}_2$ to zero.

The task function instantaneous error is represented by two-variable linguistic fuzzy sets {N, P}, respectively stand for Negative and Positive error of the three components $\tilde{\sigma}_{2x}$, $\tilde{\sigma}_{2y}$ and $\tilde{\sigma}_{2\theta}$. The robot wheels velocities are represented by four-variable linguistic fuzzy sets {VN, N, P, VP}, respectively stand for Very Negative, Negative, Positive and Very Positive magnitudes.

The inputs membership functions are triangular. For the inputs $\tilde{\sigma}_{2x}$, $\tilde{\sigma}_{2y}$, the universe of discourse is from -2E-5 to 2E-5, for $\tilde{\sigma}_{2\theta}$ it is from - π to π and for v_L and v_R , it is from -2000 to 2000.

A total of 128 rules are set, for each output, to provide the supervisory control. In this paper, each of the i^{th} fuzzy IF–THEN rule of the TS

fuzzy system is of the form shown in the following example [16]:

If $\tilde{\sigma}_{2x}$ is Negative and $\tilde{\sigma}_{2y}$ is Positive and $\tilde{\sigma}_{2\theta}$ is Positive and v_L is Positive and v_R is Very Positive, Then:

$$\begin{aligned}\lambda_{2x} &= c_1 \tilde{\sigma}_{2x} + c_2 \tilde{\sigma}_{2y} + c_3 \tilde{\sigma}_{2\theta} + c_4 v_L + c_5 v_R + c_6 \text{ and} \\ \lambda_{2y} &= c_7 \tilde{\sigma}_{2x} + c_8 \tilde{\sigma}_{2y} + c_9 \tilde{\sigma}_{2\theta} + c_{10} v_L + c_{11} v_R + c_{12} \\ \lambda_{2\theta} &= c_{13} \tilde{\sigma}_{2x} + c_{14} \tilde{\sigma}_{2y} + c_{15} \tilde{\sigma}_{2\theta} + c_{16} v_L + c_{17} v_R + c_{18}\end{aligned}$$

The tuning of the rules coefficients c1 to c18 was performed according to the fuzzy identification process described in [16].

4.2 Adaptation of matrix Λ_3 elements

The adaptation of matrix Λ_3 runs in parallel to the adaptation applied to matrix Λ_2 , using a similar TS fuzzy process. In this case, the outputs are λ_{3x} , λ_{3y} and $\lambda_{3\theta}$, whereas the inputs are the error $\tilde{\sigma}_3$ and the velocities v_L and v_R .

5. Simulations

A series of simulations were conducted to observe the performance of the proposed control formation approach. The simulations are carried out on MobileSim, programmed on Aria, both provided by Adept Mobile Robots. The virtual robots simulated are the Pioneer 3. The configuration file for the Pioneer 3 model recommends a maximum translational velocity of 2200mm/s by default. As this robot is moved by differential wheels (see Eq. 1), the left and right velocities should be less than 2200mm/s to avoid saturation of the actuator.

The shapes used for the formation control, i.e., ellipse, diamond, rounded rectangle, and star shapes, are regular closed, smooth, and simple planar curves. These shapes can be described by the generalized superellipse expressed in (11) and have time-varying parameters. The effectiveness of the proposed control is tested during the evolution of the shape switching.

In the simulations, each of the 21 robots in the group runs, in a decentralized manner, the non-holonomic fuzzy-adapted NSB behaviors described previously, considering the information of its two neighbour robots. Through the data communication channel, each configuration is shared with the other members of the group, as well as the relevant information regarding to the formation settings, i.e. desired curve parameters.

The desired trajectory of each robot is the set of all points that the robot has to travel according to the time-varying formation. As the shapes change their parameters, the robots track the desired positions on the boundary describing these trajectories.

The simulations are classified in four categories, which characteristics are described briefly as follows

S1. Random initial positions to elliptical

The group of 21 robots adopts an elliptical shape starting from initial positions set at random. The desired ellipse parameters (11) are $a = 8000\text{mm}$, $b = 10000\text{mm}$ and $m = 1$.

S2. Elliptical shapes

The group of 21 robots tracks the boundary of a time-varying ellipse, whose radii variations cause the ellipse to change from horizontal to vertical. The variation of the radii is given by:

$$\begin{aligned}a(t) &= a_i + (a_f - a_i)(1 - e^{-t/\tau}) \\ b(t) &= b_i + (b_f - b_i)(1 - e^{-t/\tau})\end{aligned}\quad (20)$$

with $\tau = 60\text{s}$, $a_i = 10\text{m}$, $a_f = 8\text{m}$, $b_i = 8\text{m}$ and $b_f = 10\text{m}$.

In Figure 4 are shown the trajectories and the final location of the robots for this simulation. Figure 5 shows the position errors.

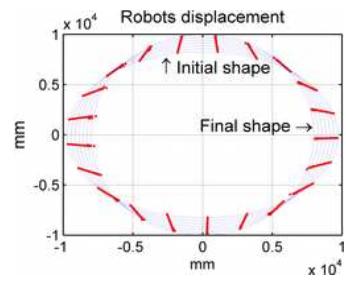


Figure 4. Trajectories and final robots location, S2.

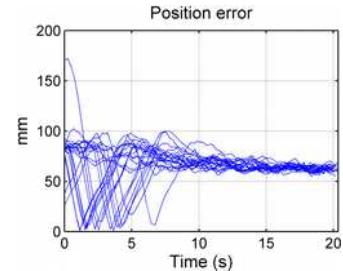


Figure 5. Position errors in S2.

In Figures 6 and 7 are shown the performances of the task functions errors, $\tilde{\sigma}_2$ and $\tilde{\sigma}_3$, which converge to zero.

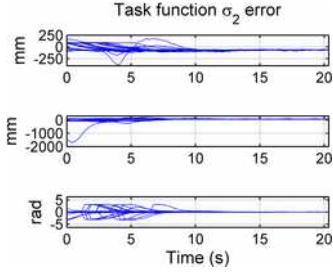


Figure 6. $\tilde{\sigma}_2$ performance in S2.

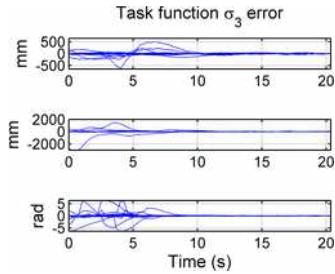


Figure 7. $\tilde{\sigma}_3$ performance in S2.

The observed left and right wheel velocities are shown in Figure 8.

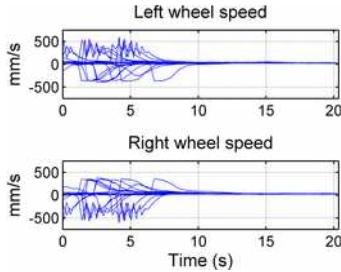


Figure 8. v_L and v_R performance in S2.

S3. Diamond to rounded rectangle

When the value of the exponent in (11) increases such that $m > 1$, the ellipse gradually adopts a diamond shape and when the exponent value decreases such that $m < 1$, the ellipse progressively turns into a rounded rectangle.

In this simulation, the shape variations are due to different values of the exponent. The radii are kept constant in $a = 8\text{m}$ and $b = 10\text{m}$. The exponent changes from the diamond conditions with $m_i = 2$, to the rounded rectangle, $m_f = 1/5$. The exponent variation is given by:

$$m(t) = m_i + (m_f - m_i)(1 - e^{-t/\tau}) \quad (21)$$

S4. Star shapes

In this simulation, the ellipse boundary takes the shape of a rounded star when its radii, $a(t)$ and $b(t)$, vary periodically in time. The radii variations can be determined as follows:

$$\begin{aligned} a(t) &= a_i + d_o \cos \beta \\ b(t) &= b_i + d_o \cos \beta \end{aligned} \quad (22)$$

where d_o is an offset value given by:

$$d_o(t) = d_{oi} + (d_{of} - d_{oi})(1 - e^{-t/\tau}) \quad (23)$$

with $\tau = 60\text{s}$, $a_i = 8\text{m}$, $b_i = 10\text{m}$, $d_{oi} = 2\text{m}$, $d_{of} = -2\text{m}$ and $0 < \beta < 2\pi$.

6. Experiments using Adept Robots

A series of experiments were conducted to observe the performance of the proposed control formation approach on real mobile robots. Three Adept pioneer mobile robots were used. The control inputs v_L and v_R , obtained from equation (8), are applied on the left and the right driving wheels. The saturation value for the velocity of the wheels is 2200mm/s, the default maximum setting.

The shapes used for the formation control in the conducted experiments are the same as in the simulations, i.e., ellipse, diamond, rounded rectangle and star shapes. The effectiveness of the proposed control is tested during the evolution of the shape switching, which is a consequence of the time-varying parameters described in equation (11).

The experiments were done in two different scenarios. First, in experiments E1 to E4, the three real robots are considered as elements of the group of 21, while the remaining 18 robots are simulated. The real robots are selected to be the first three robots in the group.

In the second scenario, experiments E5 to E8, the group consists only of the three real robots.

E1. Random initial positions to elliptical

The group of robots adopts an elliptical shape (11) starting from initial positions set at random. The desired radii are $a = 8\text{m}$, $b = 10\text{m}$ and the exponent is $m = 1$.

E2. Elliptical shapes

The group of robots tracks the boundaries of an elliptical shape while it is being modified from horizontal to vertical conditions according to (20). The initial and final values of the radii are $a_i = 10\text{m}$, $a_f = 8\text{m}$, $b_i = 8\text{m}$ and $b_f = 10\text{m}$.

E3. Diamond to rounded rectangle

The group of robots tracks the boundaries of an elliptical shape that is being modified while it changes from the diamond conditions, $m_i = 2$, to the rounded rectangle, $m_f = 1/5$ (21). The radii are kept constant in $a = 8\text{m}$ and $b = 10\text{m}$.

E4. Star shapes

The group of 3 real robots and 18 simulated robots follows the boundary of an elliptical shape with transitions due to the ellipse radii periodic variations. Then, $a(t)$ and $b(t)$ are given by (22), d_o is given by (23), with $\tau = 60\text{s}$, $a_i = 8\text{m}$, $b_i = 10\text{m}$, $d_{oi} = 2\text{m}$ and $d_{of} = -2\text{m}$.

E5. Random initial positions to elliptical, only real robots

A group of only 3 real robots adopts an elliptical shape starting from initial positions set at random. The desired elliptical shape is given by (11) with $a = 1.5\text{m}$, $b = 2\text{m}$ and $m = 1$.

E6. Elliptical shapes, only real robots

The group of 3 real robots tracks the boundary of an elliptical shape while is modified from horizontal to vertical (20). The initial and final values of the radii are $a_i = 1.5\text{m}$, $a_f = 2\text{m}$, $b_i = 2\text{m}$ and $b_f = 1.5\text{m}$. This experiment is illustrated in the following figures. Figure 9 shows the final location of the robots.



Figure 9. Final locations in E6.

Figure 10 shows the observed position errors.

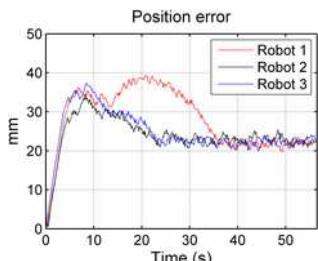


Figure 10. Position errors in E6.

Figures 11 and 12 show the performance of the task functions errors, which converge to zero.

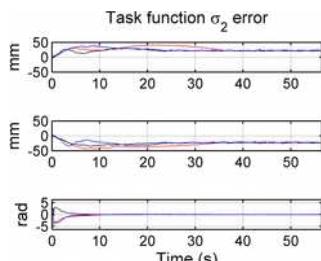


Figure 11. $\tilde{\sigma}_2$ performance in E6.

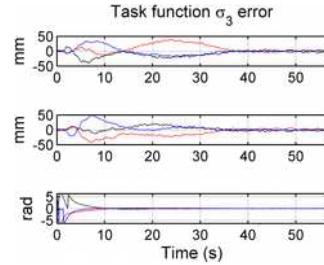


Figure 12. $\tilde{\sigma}_3$ performance in E6.

The observed left and right wheel velocities are shown in figure 13.

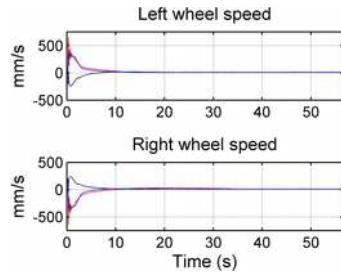


Figure 13. v_L and v_R performance in E6.

E7. Diamond to rounded rectangle, only real robots

In this experiment, 3 real robots remain on the time-varying boundary of an elliptical shape while it is modified due to different values of the exponent (11). The exponent drives the change from diamond, $m_i = 2$, to rounded rectangle conditions, $m_f = 1.5$, (21). The radii are constant in $a = 1.5\text{m}$ and $b = 2\text{m}$.

E8. Star shapes, only real robots

The 3 real robots track the boundary of changing elliptical star shapes. The periodic radii variations are given by (22), the offset d_o is given by (23) with $\tau = 60\text{s}$, $a = 1.5\text{m}$, $b = 2\text{m}$, $d_{oi} = 0.5\text{m}$ and $d_{of} = -0.5\text{m}$.

General remarks

Table 1 presents a summary of the results obtained from the simulations S1 to S4 as well as from the experimental tests E1 to E8. The data shown in the table were collected directly from the robots. As the robots move, their motion information is continuously stored (one file per robot). Table 1 shows the relevant parameters values of the 21 virtual robots (rows S1 to S4), and of the 3 real robots (rows E1 to E8). These parameters are the average values of: the minimum and maximum velocities of the left and right robot wheels, travelled distance, absolute error at the end of the transitions, and the relative error (computed as the ratio of the absolute error to the travelled distance).

Table 1. Summary of simulations and experiments results

	v_L (mm/s)		v_R (mm/s)		Travelled distance (mm)	Absolute error (mm)	Relative error %
	min	max	min	max			
S1	-982.5	1217.3	-1118.6	1295.7	14691.3	201.5	1.2
S2	-395.7	596.3	-604.1	383.7	1986.0	62.3	3.1
S3	-966.5	1679.5	-1603.7	1136.0	5859.8	56.8	1.2
S4	-470.5	905.6	-906.0	470.1	5402.3	49.1	1.5
E1	-174.8	1831.0	-287.6	1877.6	11268.7	63.0	0.7
E2	-15.0	735.0	-738.0	65.4	3239.2	64.1	2.7
E3	-460.6	688.1	-700.0	431.1	3523.1	170.4	4.8
E4	-79.4	968.2	-968.5	137.8	2830.5	84.8	3.1
E5	-1512.4	1011.1	-966.4	1023.0	11194.3	71.1	0.5
E6	-242.8	644.5	-644.6	239.1	906.4	44.7	4.9
E7	-6.6	60.4	-13.1	61.8	2839.9	187.0	7.2
E8	-430.8	738.9	-732.0	441.2	2737.9	28.2	1.1

The TS fuzzy system adapts some of the parameters of the task functions so that the left and right motor velocities are kept below the saturation values. The designed task functions drive the robots towards the desired formations. The control objectives (robot formations) are attained, in all cases, within the predetermined limits.

7. Conclusions

This paper presents a Null Space based behaviour control applied to groups of mobile robots for tracking desired trajectories by means of synchronizing the relative kinematics relationships among them, and keeping the group in formation.

One novelty of this scheme is that the Null Space based behavior control is successfully adapted to control non-holonomic differential mobile robots. The kinematics constraints are taken in consideration in the behavior aggregation that leads to the control commands. Instead of driving the robots towards specific positions, the control commands are the instantaneous velocities of the wheels.

The behaviors or the so-called task functions are designed according to the synchronization approach. The differences in the position errors, i.e. the synchronization constraints, between each pair of neighboring robots is defined as the task function in charge of leading the robot formation until the synchronization constraints goals are met. The synchronization errors

defined as the discrepancies of consecutive synchronization constraints is another measurement of the performance of the formation. For this reason there is also a task function based on the synchronization errors. These two behaviors are combined to drive both position and synchronization errors of each robot to zero during their motions. Both task functions include an orientation controller to make the robots be oriented toward their desired positions on the formation.

Another original contribution of the proposed solution is the prevention of the actuator velocity saturation by soft computing. Wheels velocities saturation can deeply affect the performance of the motion control solutions. The velocities of the wheels are the control commands generated by the Null Space for non-holonomic mobile robots. If one or more of the behavior solutions would exceed the saturation level of the wheel velocity, the whole control will be lost since a saturated task function solution annihilates the higher priority task function effects. The formation control problem in the presence of actuator velocity saturation has been addressed by means of a Takagi-Sugeno fuzzy system that adapts the elements of the matrix that directly relates the error and solution of the task functions. The Takagi-Sugeno fuzzy system allows the proposed control to correctly aggregate the different behavior solutions.

The proposed approach has been first tested by simulation on the motion control of a group of

cooperative mobile robots. The second test is experimental and its objective is to validate the approach in a more complex situation with real mobile robots achieving a three-task mission. Both simulation and experimental studies are finally performed to demonstrate the effectiveness of the proposed approach. The future work includes incorporation of the robot dynamics into the Null Space based behavior control, and a study of the performance of this control in non-structured environments.

Acknowledgments

The authors thank the personnel of the Robotics Laboratory at the Toluca Institute of Technology and to the National Council for Science and Technology of Mexico (Conacyt) for the financial support provided through its scholarship program.

REFERENCES

1. PARKER, L., **Current State of the Art in Distributed Robot Systems**, Distributed Autonomous Robotic Systems 4, Parker, L., et. al., (eds.), Springer, 2000, pp. 3-12.
2. ARAI, T., E. PAGELLO, L. PARKER, **Guest Editorial Advances in Multirobot Systems**, IEEE Trans. on Rob. and Auto., vol. 18, 2002, pp. 655-661.
3. REYNOLDS, C., **Flocks, herds, and schools: A distributed behavioral model**, Comp. Graph., vol. 21, 1987, pp. 25-34.
4. NAVARRO, I., **Formaciones de Robots**, in Proceedings of the III Simposio CEA de Control Inteligente, Spain, 2007.
5. HERNANDEZ-MARTINEZ, E. G., E. ARANDA-BRICAIRE, **Decentralized Formation Control of Multi-agent Robot Systems based on Formation Graphs**, Studies in Informatics and Control, vol. 21(1), 2012, pp. 7-16.
6. ARKIN., R. C., **Behavior-Based Robotics**, USA, MIT Press, 1998.
7. BROOKS, R. A., **A Robust Layered Control System for a Mobile Robot**, IEEE Journal of Robotics and Auto., vol. 2, 1986, pp. 14-23.
8. ARKIN, R.C., **Motor Schema - Based Mobile Robot Navigation**, The Int. J. of Rob. Research., vol. 8, 1989, pp. 92-112.
9. ANTONELLI, G., F. ARRICIELLO, S. CHIAVERINI, **The Null-space-based Behavioral Control for Autonomous Robotic Systems**, Intelligent Service Robotics, vol. 1, no. 1, 2008, pp. 27-39.
10. ANTONELLI, G., F. ARRICIELLO, S. CHIAVERINI, **Stability Analysis for the Null-Space-based Behavioral Control for Multi-robot Systems**, Proceedings of the 47th IEEE Conference on Decision and Control, Mexico, 2008.
11. SIEGWART, R., I. R. NOURBAKHSH, **Introduction to Autonomous Mobile Robots**, England, MIT Press, 2004.
12. ANTONELLI, G., F. ARRICIELLO, S. CHIAVERINI, **The NSB Control: a Behavior-based Approach for Multi-robot Systems**, Paladyn, J. of Behavioral Robotics. vol. 1, 2010, pp. 48-56.
13. ZOHAIB, M., S. M. PASHA, N. JAVAID, A. SALAAM, J. IQBAL, **An Improved Algorithm for Collision Avoidance in Environments having U and H Shaped Obstacles**, Studies in Informatics and Control, ICI Publishing House, vol. 23, no. 1, 2014, pp. 97-106
14. SUN, D., C. WANG, W. SHANG, G FENG, **A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations**, IEEE Trans. on Rob., vol. 25, no. 5, 2009, pp. 1074-1086.
15. ARRICIELLO, F., S. CHIAVERINI, G. INDIVERI, P. PEDONE, **The Null-space-based Behavioral Control for Mobile Robots with Velocity Actuator Saturation**, International Journal of Robotics Research., vol. 29, no. 10, 2010, pp. 1317-1337.
16. TAKAGI, T., M. SUGENO, **Fuzzy Identification of Systems and Its Applications to Modeling and Control**, IEEE Transactions on Systems, Man and Cybernetics, vol. 15, 1985, pp. 116-132.