Kybernetika

Alejandro Rodriguez-Angeles; Luis-Fernando Vazquez Chavez

Bio-inspired decentralized autonomous robot mobile navigation control for multi agent systems

Kybernetika, Vol. 54 (2018), No. 1, 135–154

Persistent URL: `http://dml.cz/dmlcz/147155`

Terms of use:

© Institute of Information Theory and Automation AS CR, 2018

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* `http://dml.cz`

KYBERNETIKA — VOLUME *54* (2018), NUMBER 1, PAGES 135–154

BIO-INSPIRED DECENTRALIZED AUTONOMOUS ROBOT MOBILE NAVIGATION CONTROL FOR MULTI AGENT SYSTEMS

Alejandro Rodriguez-Angeles and Luis-Fernando Vazquez-Chavez

This article proposes a decentralized navigation controller for a group of differential mobile robots that yields autonomous navigation, which allows reaching a certain desired position with a specific desired orientation, while avoiding collisions with dynamic and static obstacles. The navigation controller is constituted by two control loops, the so-called *external control* loop is based on crowd dynamics, it brings autonomous navigation properties to the system, the *internal control loop* transforms the acceleration and velocity references, given by the external loop, into the driving translational and rotational control actions to command the robots. The controller physical application could be based on several onboard sensors information, in such a way that the control strategy can be programmed individually into a group of mobile robots, this allows a decentralized performance, rendering the crowd dynamics behavior.

Each mobile robot is considered as an agent to which it is associated a *comfort zone* with a certain radius, that produces a repulsive force when it is trespassed by its environment or by another agent, this yields the necessary response to avoid collisions. Meanwhile, attractive forces drive the agents from their instantaneous position to the desired one. For collision-free navigation, Lyapunov stability method allows obtaining the stability conditions of the proposed controller and guarantees asymptotic convergence to the desired position and orientation. The navigation controller is tested by simulations, which supports the stability and convergence theoretical results.

Keywords: crowd dynamics, mobile robot, autonomous navigation

Classification: 68T40, 68T42, 93C85

1. INTRODUCTION

When developing autonomous vehicles two major issues have to be considered, first, position determination of the vehicle, second, obstacle collision avoidance. Particularly for the second case, several methods are proposed, some of them based on potential fields or geometric relations between the vehicle and its environment. Potential fields are in general based on external sensors, which involves that the environment is structured and well known, then internal sensors might be used as a complement to determine when an

DOI: 10.14736/kyb-2018-1-0135

action should be performed. A potential field is created around an obstacle, such that the distance to it determines the strength of the repulsive effect, e. g. [10].

The construction of the potential field implies knowing the dimensions of the object and its position on beforehand. In [15], potential fields are considered in combination with virtual obstacles in order to generate dynamic path-planning for an autonomous mobile robot, the results are confirmed by simulations and experiments, but lack a formal stability analysis. Usually, potential fields offer suboptimal solutions to the path planning problem, this because of the local minimum issue, as well as interactions between different potential fields. Trying to avoid local solutions, in [13] is proposed to use the so-called pseudo-bacterial genetic algorithm to guarantee an optimal and safe path for autonomous navigation, once more the goal is to obtain a safe path but knowledge of the obstacles is still required. To generate dynamic safe paths in presence of static and dynamic obstacles, in [5], is considered a heuristic-based method to search the feasible initial path efficiently, then the results are combined with an optimization algorithm for dynamic robot path planning.

In order to avoid collisions the so-called Geometric Obstacle Avoidance Control Method (GOACM), has been considered, this method is based on information regarded for onboard sensors, see for instance [4]. When one of the sensors detects an obstacle within a safety range the GOACM is triggered. This method uses the distance and the angle related to the sensor to determine a collision-free waypoint, that is commonly located on a specified distance upon a perpendicular line to the obstacle. This method is fully reactive and does not consider previous knowledge of the obstacles, however, GOACM, in general, fails for non-convex obstacles, corridors, or highly irregular obstacle shapes.

Another common way to achieve autonomous motion and obstacle avoidance for a mobile robot is by self-generating a map of the environment and classifying the obstacles that lie within it. Such techniques as the Simultaneous Localization and Mapping (SLAM) are very popular in mobile robots. In [2] a map-building technique is presented, it integrates onboard resources, the effectiveness of its method is proved by indoor experiments.

From the above-mentioned approaches, it is clear that most methods for autonomous navigation are focused on either independent navigation control or obstacle avoidance control. Such that few works addressed both problems in a unified strategy. In [12] an integrated approach to solving navigation and obstacle avoidance simultaneously is proposed, that solution is based on fuzzy logic and it is tested by simulations. Another work that integrates fuzzy logic techniques for trajectory control of mobile robots is [1], in this work an Ant Colony Optimization approach is considered in combination with fuzzy logic to render a fuzzy control for trajectory tracking, however, the safe path planning problem is led as another stage.

It is known that efficient autonomous navigation and obstacle avoidance can be found in biological systems, such as schools of fish, herds of quadrupeds, flocks of flying birds and human crowd motion. A way to model such systems is by considering the individuals as self-driven particles, see [16]. Research on crowd dynamics helps to predict large crowd behavior at a series of different events with particular circumstances, e. g. festivals or football matches as described in [6], [7], and [8]. When information regarding the movement of a crowd is well known it can be used to describe natural phenomena as mentioned above. A bio-inspired application related to mobile robots is presented in [11], where a bio-inspired collision avoidance method is proposed for a hexapod walking robot. On the other hand in [11], insect behavior is considered to design a collision avoidance algorithm based on visual information.

The study of biological navigation systems, particularly in humans, has been useful to design mobile robot obstacle avoidance strategies. It is well known that mobile robots interact with humans frequently, whereby, collisions are likely to occur. This is where lies the importance to consider how humans avoid collisions in order to design the avoiding maneuvers of the mobile robots as well. In [14] human gait characteristics are taken into account to determine the evasive actions of a robot, experimental results support this proposal, but a question arises when considering human collision avoidance behavior and mobile robot interaction, i. e. Why not to characterize human collision avoidance actions, such as in [7], and design robot collision avoidance strategies based on such human behavior?

In this article, the goal is to provide autonomy to a differential mobile robot, considered as an agent into a multi-agent system, in order to move it from an initial to a final goal position, reaching a specific desired orientation and evading dynamic and static obstacles such as other agents and environment boundaries. The proposed controller is based on the crowd dynamics model introduced by Dirk Helbing et al. [7], which describes the behavior of multiple particles that are immersed into the same environment, by the so-called *generalized force model.* This model is used to control the dynamics of a mass point (particle), in order to move it in a prescribed direction at a desired velocity.

In this approach, the above-mentioned particles are substituted by differential mobile robots, as in the base model, each one has an associated comfort zone, which when is violated by another mobile robot, wall or static obstacle, generates a repulsive force to steer the mobile robot away from it. When there is no possibility of collision, the goal position attractive forces are dominant, because of that, a straight line direction and the distance are computed from the actual instantaneous position to the desired one. Using the distance between nearest neighbors and to a final position has been widely used to design multi-agent systems where mobile robot formation is achieved, e. g. [3]. Extending the ideas from Helbing [7] to the case of differential mobile robots is not immediate, since the last one is generally modeled at a kinematic level and controlled by a combination of translational and angular velocities, whilst in [7] acceleration of the mass point is taken into account. Furthermore, when a particle is considered the orientation is worthless, in contrast, for differential mobile robots, the orientation control is fundamental to produce translational velocities that satisfy the nonholonomic constraint that this kind of robots manifests.

An extension of [7] is proposed resulting in a Bio-inspired autonomous navigation controller, so that, navigation performance resembles the behavior of an individual into a crowd situation within a constrained environment, performing evasive maneuvers while driving to the target position, where it will be oriented in a certain direction. The proposed navigation controller yields asymptotic convergence to the desired position and orientation in collision-free navigation conditions, the Lyapunov method is used to guarantee such stability properties. The above is tested by a series of simulations in a multi-agent system constituted by differential mobile robots. For comparison purposes, two cases are considered, collision-free navigation and collision navigation.

2. AUTONOMOUS NAVIGATION CONTROLLER DESIGN

The multi-agent system is conformed by differential mobile robots, that are modeled by their kinematics as in (1), this can be graphically seen in Figure 1.

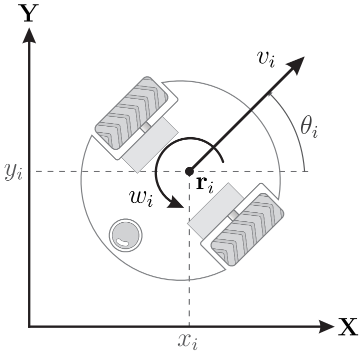

Fig. 1. Differential mobile robot.

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i \\ \dot{y}_i &= v_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i . \end{aligned} \tag{1}$$

Where the pose of each robot is given by its x_i-position, y_i-position, represented through the instantaneous position vector $\mathbf{r}_i = [x_i \; y_i]$, and the orientation angle θ_i. The robot control inputs are the translational and angular velocities, v_i and ω_i respectively, where the subindex i identifies the ith robot into the multi-agent system.

The proposed decentralized autonomous navigation controller is composed by two control loops, see Figure 2, the *external control loop* is based on crowd dynamics, it allows to achieve a desired position $[x_i \; y_i]$ and brings autonomous navigation properties including obstacle avoidance, meanwhile, the *internal control loop* transforms the acceleration and velocity references, given by the external loop, into the driving translational v_i and rotational ω_i control actions of the robot, yielding regulation to a desired orientation θ_{d_i} as well.

2.1. External control loop

This control loop processes the generalized forces Bio-inspired algorithm proposed by Dirk Helbing et al. [7], in which is assumed that the forces developed by each agent consist of both socio-psychological and physical forces.

For fully comprehension purposes the original force model is presented and described in depth, besides, this is useful to contrast the original force model to the particular one that is proposed to drive differential mobile robots. In (2) the total resulting force

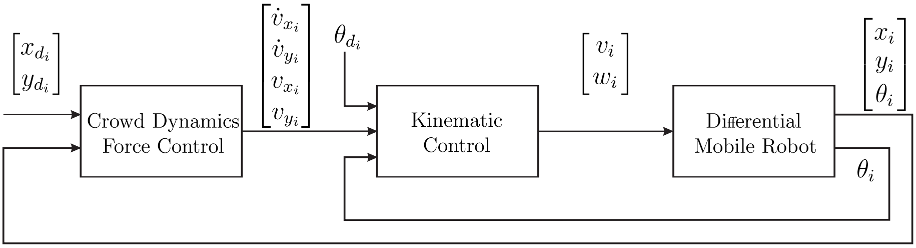
$\begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix}$
$\begin{bmatrix} \dot{v}_{x_i} \\ \dot{v}_{y_i} \\ v_{x_i} \\ v_{y_i} \end{bmatrix}$
θ_{d_i}
$\begin{bmatrix} v_i \\ w_i \end{bmatrix}$
$\begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix}$
Crowd Dynamics Force Control
Kinematic Control
Differential Mobile Robot
θ_i

Fig. 2. Closed loop navigation controller.

is calculated by adding up three different terms, the first one delivers the forces that are needed to lead the particle in a certain direction, the second term $\mathbf{f}_{ij}$ represents the forces that are manifested when interactions between particles arises, the third and final term $\mathbf{f}_{iW}$ denotes the reaction forces between ith agent and its environment or static obstacles.

$$m_i\dot{\mathbf{v}}_i(t) = m_i\frac{v_i^0(t)\mathbf{v}_{d,i}(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_{W} \mathbf{f}_{iW}. \tag{2}$$

As mentioned earlier, it is considered a group of particles of mass m_i, where their movement is triggered by attractive and repulsive forces, providing translational velocity $\mathbf{v}_i(t) = [v_{x_i}\ v_{y_i}]^T$ and acceleration $\dot{\mathbf{v}}_i(t) = [\dot{v}_{x_i}\ \dot{v}_{y_i}]^T$ references in X and Y direction. Each particle is trying to adjust its actual velocity $\mathbf{v}_i(t)$, in a certain characteristic time τ_i, to the desired velocity composed by $v_i^0(t)\mathbf{v}_{d,i}(t)$ as long as the ith agent does not interact with other agents or obstacles.

The term $\mathbf{f}_{ij}$ in (2), generates repulsive interaction forces to avoid collisions between agents i and j as follows.

$$\mathbf{f}_{ij} = \left[A_i e^{[(r_{ij}-d_{ij})/B_i]} + k_i g_i(r_{ij} - d_{ij})\right] \mathbf{n}_{ij} + \kappa_i g(r_{ij} - d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij}. \tag{3}$$

Where $A_i e^{[(r_{ij}-d_{ij})/B_i]}$ describes a tendency to remain into a personal space, reproducing the behavior of humans to stay away, as far as they feel comfortable, from other people, here A_i and B_i are constant gains. $r_{ij} = (r_i + r_j)$ represents the sum of the radii of two agents comfort zones, the distance between two agents is denoted by $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$, with $\mathbf{r}_i$ and $\mathbf{r}_j$ being their instantaneous positions, the normalized vector pointing from agent j to agent i is $\mathbf{n}_{ij} = [n_{ij,x}\ n_{ij,y}] = (\mathbf{r}_i - \mathbf{r}_j)/d_{ij}$.

The terms $k_i g(r_{ij} - d_{ij})\mathbf{n}_{ij}$ and $\kappa_i g(r_{ij} - d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij}$ render normal and tangential forces respectively, such forces arise when agents invade another agent's comfort zone. The normal and tangential forces role is to push, slow down and turn the agents, depending on the case, in order to stay away from other agents and obstacles. Function $g(\cdot)$ is zero if $d_{ij} \geq r_{ij}$, this is when the comfort zones aren't violated, otherwise i. e. when $d_{ij} < r_{ij}$, $g(\cdot)$ is equal to its argument, in this case $(r_{ij} - d_{ij})$. The tangential vector is given by $\mathbf{t}_{ij} = (-n_{ij,y}, n_{ij,x})$, while the tangential relative velocity between agents can be expressed as $\Delta v_{ji}^t = (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{t}_{ij}$.

On the other hand, the reaction forces $\mathbf{f}_{iW}$ between agents and its environment or static obstacles depend on the approaching velocity of ith agent to the obstacle W, which is given by

$$\mathbf{f}_{iW} = \left[A_i e^{[(r_i - d_{iW})/B_i]} + k_i g_i(r_i - d_{iW})\right] \mathbf{n}_{iW} + \kappa_i g(r_i - d_{iW})(\mathbf{v}_i \cdot \mathbf{t}_{iW})\mathbf{t}_{iW}. \tag{4}$$

As in (3), the term $A_i e^{[(r_i - d_{iW})/B_i]}$ describes the tendency to keep free their personal space and to stay away from obstacles in the environment. r_i is the radii of the ith agent comfort zone, d_{iW} is the distance between the agent and a static obstacle W, while $\mathbf{n}_{iW}$ is the normalized vector between the agent i and the obstacle W, finally, $\mathbf{t}_{iW}$ is the normalized tangential vector. Same as for the interaction forces, the function $g(\cdot)$ becomes zero when there are no interactions and its argument otherwise.

In equations (3) and (4), k_i and κ_i are positive constant parameters that can be changed to influence the magnitude of the normal and tangential forces respectively.

Note that in the repulsive force expressions (3) and (4) the normal component depends only on the distance to the obstacle, meanwhile the tangential terms, that causes rotation of the agents, take into account whether the obstacle is dynamic or static, particularly in (3) the tangential term considers the relative approaching velocity between agents, whereby, the higher the relative approaching velocity, harder the turning action. On the other side in (4), is only considered the ith agent velocity which corresponds to the approaching velocity to the obstacle. Through this, it becomes evident that the difference within that expressions resides in the way which the relative speed between agents and obstacles is considered, therefore, when an agent stops moving it will be recognized as a static obstacle by the rest of the agents.

In contrast to the Helbing's force model, at our multi-agent navigation system it is required to move each agent from its instantaneous position $\mathbf{r}_i = [x_i \; y_i]^T$ to the desired one $\mathbf{r}_{d_i} = [x_{d_i} \; y_{d_i}]^T$, whereby the original force model is modified considering distance based control techniques, thus the ith robot position error $\mathbf{e}_i$ is defined as

$$\mathbf{e}_i = \begin{bmatrix} e_{x,i} \\ e_{y,i} \end{bmatrix} = \begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{5}$$

Equation (5) provides a vector through which the ith agent could reach its objective, simultaneously it calculates the distance between the instantaneous and final positions, thus the agent will move faster as farther from the desired position, and slower as closer to it. This might result in unsatisfactory performance because of slow convergence, [17]. To render a better response and faster convergence, normalization of the vector $\mathbf{e}_i$ is considered.

Thereby each agent will try to reach its desired position following a straight line calculated from the vector $\mathbf{e}_i$, whose normalization allows the desired velocity magnitude v_i^0, which is considered as a constant, to be imposed during the trajectory towards the desired position. The agent will stop moving when the final position is achieved i. e. when $\mathbf{e}_i = 0$. Since the mobile robots are kinematically controlled their masses could be considered unitary, thus the proposed force model is given by

$$\dot{\mathbf{v}}_i(t) = \frac{v_i^0 \mathbf{e}_i(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_W \mathbf{f}_{iW}. \tag{6}$$

Once more the first term in (6) causes that each agent regulates its actual velocity $\mathbf{v}_i(t)$ attempting to equalize the magnitude of the desired speed v_i^0 in a certain characteristic time τ_i, which represents the time gap in that the comparison and adjustment of the velocity variables are carried out, at the same time, the agent is steering in the desired direction $\mathbf{e}_i$, meanwhile the two last terms $\mathbf{f}_{ij}$ and $\mathbf{f}_{iW}$ are intended to generate repulsive forces to avoid collisions with dynamic and static obstacles respectively.

From analysis and simulations of the repulsive force components (3) and (4), it is concluded that the terms $A_i e^{[(r_{ij}-d_{ij})/B_i]}$ and $A_i e^{[(r_i-d_{iW})/B_i]}$, generate a permanent repulsive effect between agents and from agents to static obstacles respectively, regardless of whether the agents reach the desired position or not, this situation avoids the possibility to maintain the final position once it has been reached. For this reason, in the proposed navigation controller these terms are discarded. Thus, the interaction forces between agents are given by

$$\mathbf{f}_{ij} = k_i g(r_{ij} - d_{ij})\mathbf{n}_{ij} + \kappa_i g(r_{ij} - d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij}. \tag{7}$$

Where each variable has already been described. By performing the same modification as in (7), the reaction forces between agents and its environment are given by

$$\mathbf{f}_{iW} = k_i g(r_i - d_{iW})\mathbf{n}_{iW} + \kappa_i g(r_i - d_{iW})(\mathbf{v}_i \cdot \mathbf{t}_{iW})\mathbf{t}_{iW}. \tag{8}$$

Where all the involved variables were described above.

2.2. Internal control loop

This control loop relates the force model signals v_{x_i} and v_{y_i}, (components of vector $\mathbf{v}_i$) and the acceleration ones $\dot{v}_{x_i}$ and $\dot{v}_{y_i}$ given by (6), into the driving signals v_i and ω_i of the kinematic model (1). So the translational velocity input v_i is proposed as:

$$v_i = K_{v_i}\left(v_{x_i}\cos(\theta_i) + v_{y_i}\sin(\theta_i)\right) \tag{9}$$

with K_{v_i} a positive constant tuning gain.

Note that the above velocity control action tries to generate an appropriate translational velocity in order to achieve the desired cartesian position, however, it is still required to generate the appropriate rotational velocity profile. Furthermore, it is not only important to achieve the cartesian goal position but some tasks do require to reach certain desired orientation value.

Since R. W. Brocket [18] has established that a non-holonomic system can only be controlled by discontinuous or time-variant control laws, in order to guarantee that the proposed controller overcomes this restriction, the structure for the rotation control variable is based on non-linear time-varying feedback control laws, that was developed by C. Samson et al. [19, 20] as well as by C. Canudas et al. [21], that allows the mobile robot to be controlled from its center of rotation which is located at the midpoint of the wheel axis. This ensures a complete control of the agents even while their non-holonomic constraints are considered, moreover, it becomes possible to regulate their orientation by introducing the definition of the orientation error e_{θ_i}, thus the expression (10) calculates the difference between the instantaneous agent orientation θ_i and the desired one θ_{d_i}.

$$e_{\theta_i} = \theta_i - \theta_{d_i}. \tag{10}$$

So that the orientation control action is proposed as

$$\omega_i = K_{w_i}\left(\frac{\dot{v}_{y_i}v_{x_i} - \dot{v}_{x_i}v_{y_i}}{\epsilon + v_i^2}\right)\frac{\sin(e_{\theta_i})}{e_{\theta_i}} - K_{\theta_i}e_{\theta_i}. \tag{11}$$

Where K_{w_i} and K_{θ_i} are positive, constant control gains, also a positive parameter $\epsilon \approx 0$ is introduced to avoid the singularity when $v_i = 0$.

Note that (11) is well defined since $\frac{\sin(e_{\theta_i})}{e_{\theta_i}}$ is a continuous smooth function in e_{θ_i}, which satisfies that $\frac{\sin(e_{\theta_i})}{e_{\theta_i}} = \int_0^1 \cos(\rho e_{\theta_i})\,\mathrm{d}\rho$ and $\lim_{e_{\theta_i}\to 0}\frac{\sin(e_{\theta_i})}{e_{\theta_i}} = 1$.

Based on the above considerations, the navigation controller will allow each agent to reach the desired position references, orientate themselves to the required value and stay in position at the end of the trajectory.

3. OBSTACLE FREE NAVIGATION: CLOSED LOOP STABILITY

In this section stability properties for obstacle-free navigation are provided, specifically the closed loop between controller (6), (7), (8), (9), (11) and the mobile robot described by (1). When obstacle-free navigation is considered the repulsive forces (7) and (8) are zero. Thus, by defining the state variables $z_{1,i} = e_{x,i}$, $z_{2,i} = e_{y,i}$, $z_{3,i} = e_{\theta_i}$, $z_{4,i} = v_{x_i}$ and $z_{5,i} = v_{y_i}$, the closed loop for the ith agent is given by

$$\dot{\mathbf{z}}_i = \begin{bmatrix}\dot{z}_{1,i}\\ \dot{z}_{2,i}\\ \dot{z}_{3,i}\\ \dot{z}_{4,i}\\ \dot{z}_{5,i}\end{bmatrix} = \begin{bmatrix} -K_{v_i}\left(z_{4,i}\cos^2(\theta_i) + z_{5,i}\sin(\theta_i)\cos(\theta_i)\right)\\ -K_{v_i}\left(z_{4,i}\sin(\theta_i)\cos(\theta_i) + z_{5,i}\sin^2(\theta_i)\right)\\ \frac{K_{w_i}v_i^0}{\tau_i(\epsilon + v_i^2)}\left(z_{2,i}z_{4,i} - z_{1,i}z_{5,i}\right)\frac{\sin(z_{3,i})}{z_{3,i}} - K_{\theta_i}z_{3,i}\\ \frac{1}{\tau_i}\left(v_i^0 z_{1,i} - z_{4,i}\right)\\ \frac{1}{\tau_i}\left(v_i^0 z_{2,i} - z_{5,i}\right)\end{bmatrix}. \tag{12}$$

Note that the origin $\dot{\mathbf{z}}_i = 0$ is an equilibrium point of the closed loop system (12), such that stability properties can be stated as follows.

Theorem 3.1. Considering the collision-free navigation case, such that the closed loop between controller (6), (7), (8), (9), (11) and the ith mobile robot (1) is given by (12). If all control gains are positive and the conditions imposed to the control gains given by (13) are satisfied, then the origin of the closed loop system is asymptotically stable.

$$0 < \frac{K_{w_i}}{\epsilon} < 1\ ,\ \frac{v_i^0}{\tau_i} > \frac{1.2K_{v_i}}{1 - \frac{K_{w_i}}{\epsilon}}. \tag{13}$$

Proof. Consider the candidate Lyapunov function $V_i = \frac{1}{2}\mathbf{z}_i^T\mathbf{z}_i$ that is a positive definite function. By straight forward computations the time derivative of V_i is given by

$$\begin{aligned} \dot{V}_i &= -K_{\theta_i} z_{3,i}^2 - \frac{z_{4,i}^2}{\tau_i} - \frac{z_{5,i}^2}{\tau_i} - z_{1,i} z_{4,i} \left\{ -\frac{v_i^0}{\tau_i} + K_{v_i} \cos^2(\theta_i) \right\} \\ & \quad -z_{1,i} z_{5,i} \left\{ \frac{K_{w_i} v_i^0}{\tau_i(\epsilon + v_i^2)} \sin(z_{3,i}) + K_{v_i} \sin(\theta_i)\cos(\theta_i) \right\} \\ & \quad -z_{2,i} z_{4,i} \left\{ \frac{-K_{w_i} v_i^0}{\tau_i(\epsilon + v_i^2)} \sin(z_{3,i}) + K_{v_i} \sin(\theta_i)\cos(\theta_i) \right\} \\ & \quad -z_{2,i} z_{5,i} \left\{ -\frac{v_i^0}{\tau_i} + K_{v_i} \sin^2(\theta_i) \right\}. \end{aligned} \tag{14}$$

In order to proof that $\dot{V}_i$ is definite negative, its upper bound is computed, for that it is considered that $\frac{1}{2}\left(a^2 + b^2\right) \geq -ab$, furthermore since $v_i^2 \geq 0$ then, worst case implies $v_i^2 = 0$, such that $\dot{V}_i$ can be rewritten as

$$\begin{aligned} \dot{V}_i &\leq -K_{\theta_i} z_{3,i}^2 - \frac{z_{4,i}^2}{\tau_i} - \frac{z_{5,i}^2}{\tau_i} \\ & \quad -z_{1,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 - \frac{K_{w_i}}{\epsilon} \sin(z_{3,i}) \right] - K_{v_i} \left[\cos^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \right] \right\} \\ & \quad -z_{2,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 + \frac{K_{w_i}}{\epsilon} \sin(z_{3,i}) \right] - K_{v_i} \left[\sin^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \right] \right\} \\ & \quad -z_{4,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 + \frac{K_{w_i}}{\epsilon} \sin(z_{3,i}) \right] - K_{v_i} \left[\cos^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \right] \right\} \\ & \quad -z_{5,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 - \frac{K_{w_i}}{\epsilon} \sin(z_{3,i}) \right] - K_{v_i} \left[\sin^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \right] \right\}. \end{aligned} \tag{15}$$

By considering the trigonometric functions in (15), it follows that $-1 \leq \sin(z_{3,i}) \leq 1$ for $z_{3,i} \in \Re$, and $-0.2 \leq \cos^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \leq 1.2$, $-0.2 \leq \sin^2(\theta_i) + \sin(\theta_i)\cos(\theta_i) \leq 1.2$ for $\theta_i \in \Re$, therefore by considering worst case, (15) can be bounded as

$$\begin{aligned} \dot{V}_i &\leq -K_{\theta_i} z_{3,i}^2 - \frac{z_{4,i}^2}{\tau_i} - \frac{z_{5,i}^2}{\tau_i} \\ & \quad -z_{1,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 - \frac{K_{w_i}}{\epsilon} \right] - 1.2 K_{v_i} \right\} - z_{2,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 + \frac{K_{w_i}}{\epsilon} \right] - 1.2 K_{v_i} \right\} \\ & \quad -z_{4,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 + \frac{K_{w_i}}{\epsilon} \right] - 1.2 K_{v_i} \right\} - z_{5,i}^2 \left\{ \frac{v_i^0}{\tau_i} \left[1 - \frac{K_{w_i}}{\epsilon} \right] - 1.2 K_{v_i} \right\}. \end{aligned} \tag{16}$$

The conditions given by (13) would guarantee that $\dot{V}_i$ is definite negative and $\dot{V}_i = 0$ only at the origin, thus, from standard Lyapunov theory [9], it can be concluded that the origin of the closed loop system (12) is asymptotically stable. □

Remark 3.2. Stability analysis of the cases where collisions with dynamic and static obstacles are possible is not carried out, this because the reaction forces could be different for each collision situation. For that reason, when the possibility of collisions arises, it has to be guaranteed that the control gains k_i and κ_i, in the repulsive components (7) and (8) will result in forces enough to generate fast turning of the agents and therefore avoiding collisions.

4. SIMULATION RESULTS

In order to prove the performance of the proposed controller over a multi-agent system, six differential mobile robots are considered.

A simulator was made in MATLAB®, it creates a movie showing static obstacles and the movement of each agent as well as the trajectory followed to avoid collisions, see Figure 3. The blue lines and circular shape represent the environment which, in this case, is given by horizontal walls at 0 and 1.1 [m] and a column of radius 0.25 [m], in the middle. The goal is to move the agents from the initial position at the left to the desired position at the right, where every agent must reach a desired final orientation. The agents are represented by the small circles whit an inscribed pentagon pointing as its orientation angle. The bigger circle around each agent represents their comfort zone, which allows identifying interactions and possible collisions.

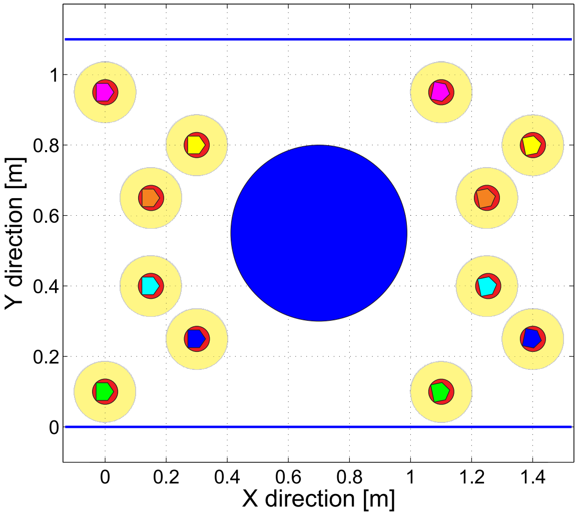

Fig. 3. Simulation enviroment.

To carry out the tests, two cases are simulated, in the first one, a collision-free environment is considered i. e. there is no interaction between agents and obstacles. In the second case, a circular object (column) is introduced in the center of the working area as a static obstacle, this generates the possibility of collisions between agents since each one will try to evade the column. For both cases, free and possible collision navigation, the initial conditions, desired position, and orientation remains the same for every agent, these values are listed in Table 4. Along with all simulations normalization of $\mathbf{e}_i$, given by (5), is considered to yield faster convergence.

Agent	$x_i(0)$	$y_i(0)$	$\theta_i(0)$	x_{d_i}	y_{d_i}	θ_{d_i}
1	0.0	0.1	0.0	1.1	0.1	0.2
2	0.3	0.25	0.0	1.4	0.25	-0.2
3	0.15	0.4	0.0	1.25	0.4	0.2
4	0.0	0.95	0.0	1.1	0.95	-0.2
5	0.3	0.8	0.0	1.4	0.8	0.2
6	0.15	0.65	0.0	1.25	0.65	0.2

Tab. 1. Initial conditions, desired position and orientation, in [m] and [rad] respectively.

4.1. Collision-free navigation case

For this case, there is no column into the environment and due the initial and desired position and orientation values, the agents move in a straight line through the working area. The control gains are tuned in the same way for every agent as can be seen at Table 4.1. The comfort zone radius is $r_i = 0.1$ [m] and the parameter to avoid singularities in (11) is set as $\epsilon = 0.01$.

Gain	v_i^0 [m/s]	τ_i	k_i	κ_i	K_{v_i}	K_{w_i}	K_{θ_i}
Value	0.5	0.005	150	300	0.07	0.009	0.1

Tab. 2. Navigation control gains, obstacle free navigation scenery.

As expected the agents move in a straight line from its initial position to the desired one as shown in Figure 4. The cartesian position and orientation variables for each agent are shown in Figures 5, 6, and 7 respectively.

In order to analyze Figure 5 it is important to notice that from Table 4, there are three pairs of agents that share the initial and desired x-position, these pairs are agents 1 and 4, agents 2 and 5, and agents 3 and 6. Furthermore, the agents are commanded to move the same distance along the X axis and due they have the same control gains, their trajectories show the same profile shifted accordingly to the initial y-position.

Since the initial and desired y-position has exactly the same value, as it can be seen in Table 4, every agent will move through a horizontal straight line, this is clearly observed in Figure 6.

For this simulation, every agent starts with the same horizontal orientation, as shown by their initial conditions, nevertheless, there are two different desired values for orientation, there are 0.2 [rad] for agents 1, 3, 5, 6, and -0.2 [rad] for agents 2 and 4, because

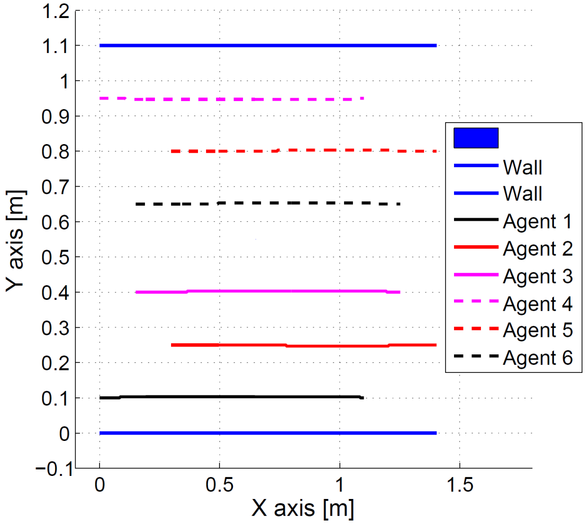

Fig. 4. Agents trajectory at the Cartesian plane, obstacle free navigation scenery.

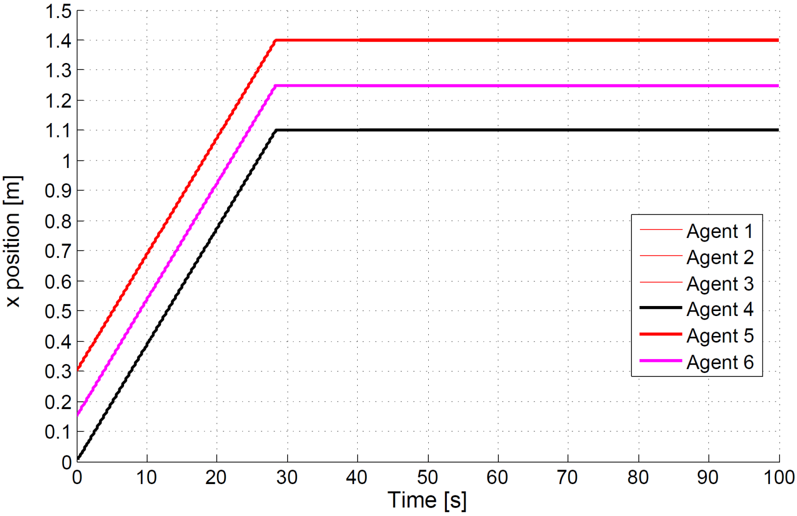

Fig. 5. Agent x-position, obstacle free navigation scenery.

all agents have the same control gains their orientation trajectories show the same profile depending on its desired final orientation, that's why there are only two trajectories displayed at the figure.

Finally the kinematic translational and rotational control actions for each agent are

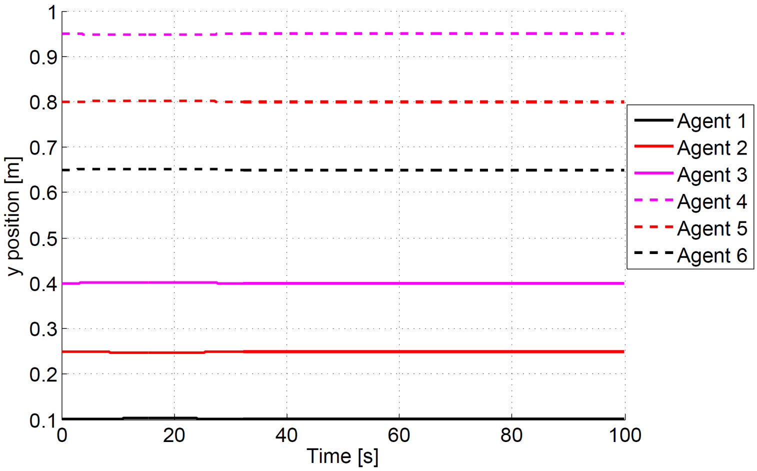

Fig. 6. Agent y-position, obstacle free navigation scenery.

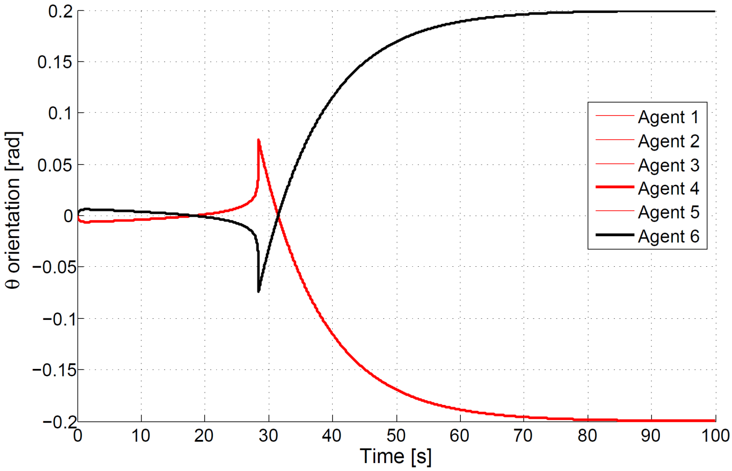

Fig. 7. Agent θ-orientation, obstacle free navigation scenery.

shown in Figures 8 and 9 respectively. Note that the same grouping of agents, as for x and y trajectories appears. All robots perform the same translational velocity because they have to move the same distance in the same amount of time. Also, it is possible to observe that agents 1, 3, 5 and 6 present the same rotational control action, as well as agents 2 and 4.

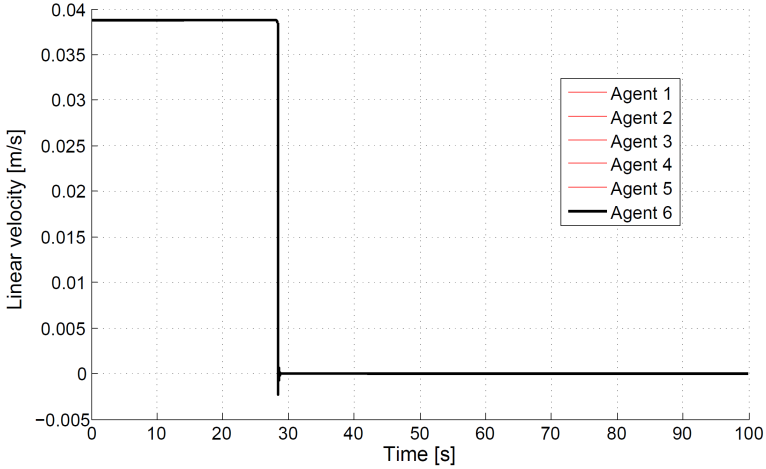

Fig. 8. Agent translational control action, obstacle free navigation scenery.

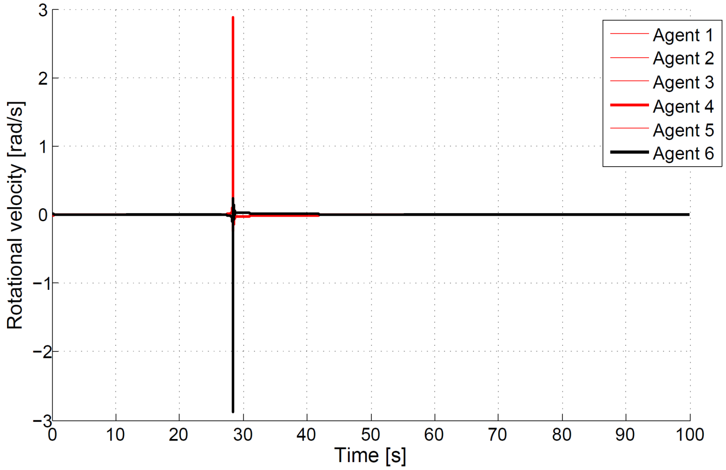

Fig. 9. Agent rotational control action, obstacle free navigation scenery.

4.2. Collision navigation case

For this case, the column is located at the center of the environment, because of this, when the agents move in a straight line to their target position, the possibility of having collisions appears due that static obstacle gets into their comfort zone. Once the agents are avoiding collisions with the column, they begin to invade another agent's space which might cause collisions between them, so the dynamic evasion component appears. By running this simulation, it was observed that not every control gain listed in Table 4.1 are suitable to avoid collisions, for this reason some gains are tunned as follows $K_{v_i} = 0.5$, $K_{w_i} = 0.09$ and $\tau_i = 0.05$.

Remark 4.1. It should be considered that the control gains in Table 4.1 were selected for obstacle-free navigation and to fulfill the stability conditions given by (13), besides the fact that such conditions were obtained by boundedness of the time derivative of the Lyapunov function, thus being conservative conditions.

The cartesian trajectory of each agent is shown in Figure 10, it can be compared with the displayed in Figure 4, it is clear that the robots modified their straight-line trajectory to avoid possible collisions with the column, nevertheless, they reach their desired position. Although it seems that the trajectories for agents 5 and 6 intersect, there is no collision, because when considering the time-based trajectories in Figure 11, at the same instant of time their x-position is completely different.

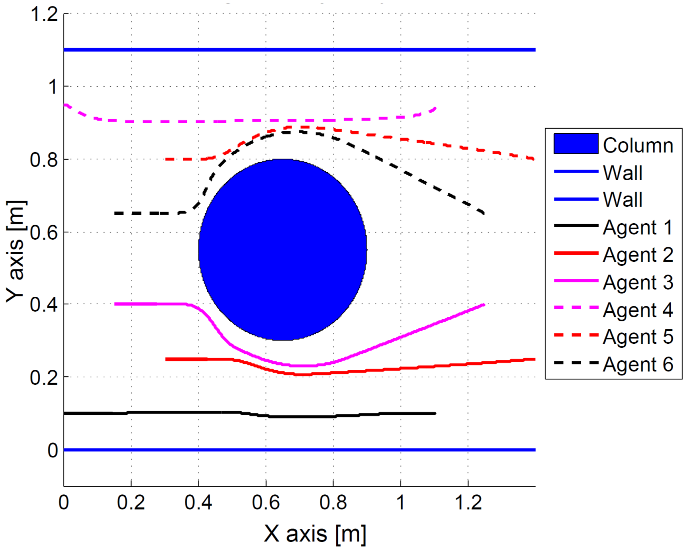

Fig. 10. Agents trajectory at the Cartesian plane, obstacle collision navigation scenery.

It can be seen in the Figure 11 that the same clusters of agents as for Figure 5 are generated, this is because they are based on identical initial and desired conditions, however, their behavior drastically changes to avoid possible collisions, which is evident for agent 6 that slows down with respect to agent 3 in order to avoid such collisions.

Same as the collision-free case, the agents are intended to go in a straight line from their initial to the desired y-position, but from Figure 12 it is evident that the agents modified their trajectory to evade the column, nevertheless, once collisions are avoided they return to move in a straight line from their instantaneous position to their desired one, which as mentioned before, is reached by every agent.

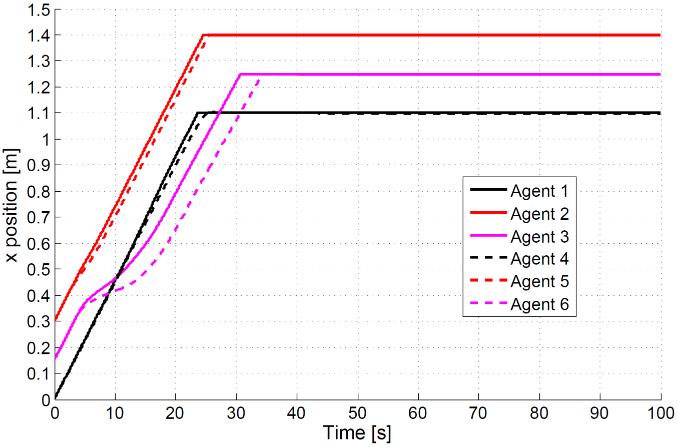

Fig. 11. Agent x-position, obstacle collision navigation scenery.

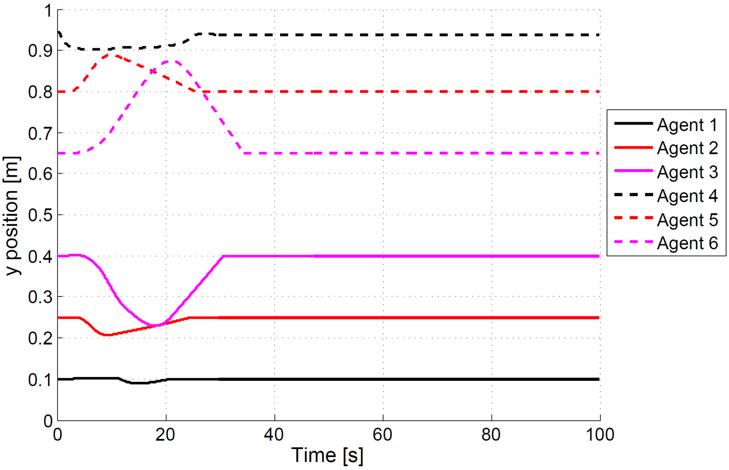

Fig. 12. Agent y-position, obstacle collision free navigation scenery.

Note at Figure 13 that all agents reached their desired orientation, but along the way, while there is still a possibility of collision, they perform avoiding maneuvers by turning, thus when the final position is achieved the controller begins to orientate the robot.

Finally the kinematic translational and rotational control actions for each agent are shown in Figures 14 and 15 respectively. Note that the agent's cluster phenomena observed in Figure 8 and 9 is lost in order to generate the control actions to avoid collisions. By now it is clear that the followed strategy is implemented by modifying the agent's translational velocity and by turning.

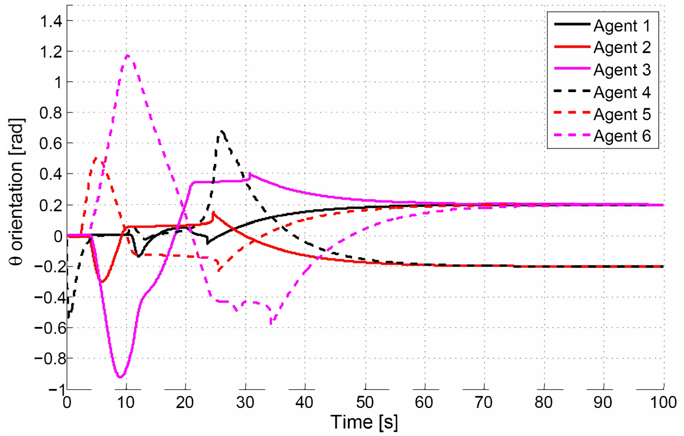

Fig. 13. Agent θ-orientation, obstacle collision navigation scenery.

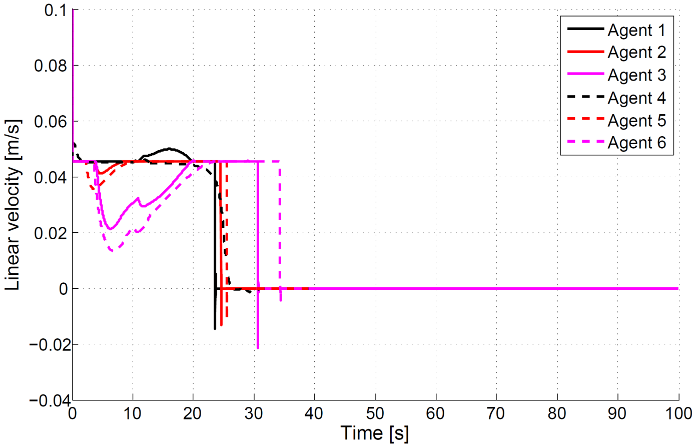

Fig. 14. Agent translational control action, obstacle collision navigation scenery.

5. CONCLUSIONS

A force based autonomous navigation controller that steers robots from their actual to a goal position, reaching certain desired orientation, has been designed. The results of applying the proposed navigation controller happen to be a full analogon of what happens in crowd dynamics. Simulations show that violation of comfort zones leads to very fast turning behavior, instead of moving directly in the direction of the repulsive force. This turning reaction can be modulated by weighting the normal and tangential terms of the repulsive force components.

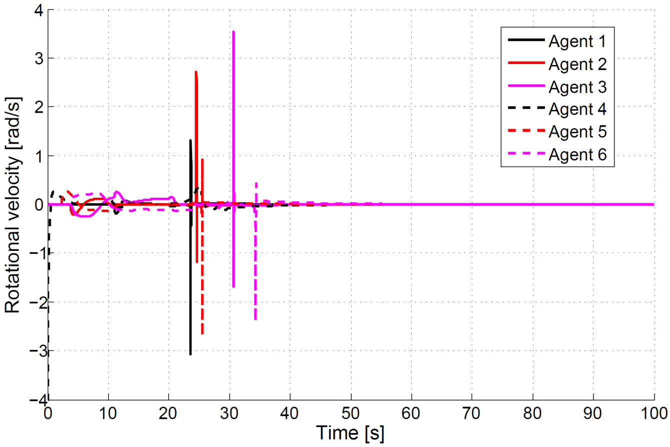

Fig. 15. Agent rotational control action, obstacle collision navigation scenery.

A formal stability analysis proves that the navigation controller yields asymptotic convergence to the desired position and orientation as long as some conditions are satisfied and no collisions are present. This result is very useful in order to tune the control gains for the collision-free case, and then adjusting their values for considering possible collisions.

The physical implementation of the navigation controller could be based on a series of onboard sensors in order to determine the distance vectors between agents and obstacles and it is even possible to calculate their relative velocities, thus being very suitable for embedded platforms. Furthermore, the decentralized architecture of the control allows to program it in several mobile robots, rendering a decentralized autonomous navigation controller that perfectly fits for multi-agent and collaborative mobile robot systems.

As future work, differential mobile robots will be built to prove the navigation controller experimentally and in real time. Also, relaxation on the upper and lower bounds of the stability conditions given by (13) is being explored, since these conditions are rather conservative.

ACKNOWLEDGEMENT

This work was partially supported by the National Council for Science and Technology, CONACyT – Mexico, under the grant 254329.

(Received June 1, 2017)

REFERENCES

[1] O. Castillo, H. Neyoy, J. Soria, P. Melin, and F. Valdez: A new approach for dynamic fuzzy logic parameter tuning in AntColony Optimization and its application

in fuzzy control of a mobile robot. Applied Soft Computing *28* (2015), 150–159. DOI:10.1016/j.asoc.2014.12.002

[2] H. Cheng, H. Chen, and Y. Liu: Topological indoor localization and navigation for autonomous mobile robots. IEEE Trans. Autonom. Sci. Engrg. *12* (2015), 2, 729–738. DOI:10.1109/tase.2014.2351814

[3] X. Chen, Y. Yang, S. Cai, and J. Chen: Modeling and analysis of multi-agent coordination using nearest neighbor rules informatics in control. In: International Asian Conference on Automation and Robotics, CAR09, 2009, pp. 273–277. DOI:10.1109/car.2009.18

[4] Y. Dai and S. G. Lee: Formation control of mobile robots with obstacle avoidance based on GOACM using onboard sensors. Int. J. Control, Automat. Systems *12* (2014), 5, 1077–1089. DOI:10.1109/car.2009.18

[5] M. S. Ganeshmurthy and G. R. Suresh: Path planning algorithm for autonomous mobile robot in dynamic environment. In: 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), 2015. DOI:10.1109/icscn.2015.7219901

[6] D. Helbing, L. Buzna, A. Johansson, and T. Werner: Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. Transport. Sci. *39* (2005), 1, 1–24. DOI:10.1287/trsc.1040.0108

[7] D. Helbing, I. Farkas, and T. Vicsek: Simulating dynamical features of escape panic. Nature *407* (2000), 487–490. DOI:10.1038/35035023

[8] D. Helbing, P. Molnár, I. J. Farkas, and K. Bolay: Self-organizing pedestrian movement Environment and planning B: planning and design. SAGE Publications Sage UK: London, England *28* (2001), 361–383. DOI:10.1068/b2697

[9] H. K. Khalil: Nonlinear Systems. Prentice Hall, Upper Saddle River 1996.

[10] D. Kostic, S. Adinandra, J. Caarls, N. van de Wouw, and H. Nijmeijer: Collision-free tracking control of unicycle mobile robots. In: Proc. 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC), 2009, pp. 5667–5672. DOI:10.1109/cdc.2009.5400088

[11] H. G. Meyer, O. J. N. Bertrand, and J. Paskarbeit: A bio-inspired model for visual collision avoidance on a hexapod walking robot. In: Biomimetic and Biohybrid Systems: 5th International Conference,Living Machines (F. Nathan, F. Lepora, A- Mura, M. Mangan, P. F. M J. Verschure, M. Desmulliez, and T. J. Prescott, eds.), Springer Verlag 2016, pp. 167–178. DOI:10.1007/978-3-319-42417-0_16

[12] H. Omrane, M. S. Masmoudi, and M. Masmoudi: Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation. Computational Intelligence and Neuroscience, 2016. DOI:10.1155/2016/9548482

[13] U. Orozco-Rosas, O. Montiel, and R. Sepulveda: Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation. Int. J. Advanced Robotic Systems *12* (2015), 7, 81. DOI:10.5772/60715

[14] T. Tomizawa and Y. Shibata: Oncoming human avoidance for autonomous mobile robots based on gait characteristics. J. Robotics Mecatronics *28* (2016), 4, 500–507. DOI:10.20965/jrm.2016.p0500

[15] K. Ueno, T. Kinoshita, K. Kobayashi, and K. Watanabe: Development of a robust path-planning algorithm using virtual obstacles for an autonomous mobile robot. J. Robotics Mechatronics *27* (2015), 3, 286–292. DOI:10.20965/jrm.2015.p0286

[16] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet: Novel type of phase transition in a system of self-driven particles. Physical Rev. Lett. *75* (1995), 6, 1226–1229. DOI:10.1103/physrevlett.75.1226

[17] C. Xuesong, Y. Yimin, C. Shuting, and C. Jianping: Modeling and analysis of multi-agent coordination using nearest neighbor rules. In: Int. Asia Conference on Informatics in Control, Automation and Robotics 2009. DOI:10.1109/car.2009.18

[18] R. W. Brockett (EDS.): Asymptotic stability and feedback stabilization. In: Differential Geometric Control Theory *27* (1983), Birkhauser, Boston 1983.

[19] C. Samson and K. Ait-Abderrahim: Feedback control of a nonholonomic wheeled cart in cartesian space. In: Proc. IEEE International Conference on Robotics and Automation 1991. DOI:10.1109/robot.1991.131748

[20] C. Samson and K. Ait-Abderrahim: Mobile Robot Control. Part 1: Feedback Control of Nonholonomic Wheeled Cart in Cartesian Space. INRIA, 1990.

[21] C. C. de Wit, B. Siciliano, and G. Bastin (eds.): Theory of Robot Control. Springer Science and Business Media 2012. DOI:10.1007/978-1-4471-1501-4

Alejandro Rodriguez-Angeles, Cinvestav-IPN, Electrical Engineering Department, Mechatronics Group, Av. I.P.N. 2508, Mexico City, C.P. 07360. Mexico.
e-mail: aangeles@cinvestav.mx

Luis-Fernando Vazquez Chavez, Cinvestav-IPN, Electrical Engineering Department, Mechatronics Group, Av. I.P.N. 2508, Mexico City, C.P. 07360. Mexico.
e-mail: fernando.vazquez@cinvestav.mx