A bio-inspired autonomous navigation controller for differential mobile robots based on crowd dynamics

Nijmeijer, H.; Rodriguez Angeles, A.; van Kuijk, F.J.M.

Published in:
Advances in Swarm intelligence

DOI:
10.1007/978-3-319-41009-8_60

Published: 01/01/2016

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

A Bio-inspired Autonomous Navigation Controller for Differential Mobile Robots Based on Crowd Dynamics

Alejandro Rodriguez-Angeles[1,2(✉)], Henk Nijmeijer[2], and Fransis J.M. van Kuijk[2]

[1] Mechatronics Section, Electrical Engineering Department, Cinvestav-IPN, Mexico City, Mexico
aangeles@cinvestav.mx, h.nijmeijer@tue.nl, f.j.m.v.kuijk@student.tue.nl
[2] Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
http://www.meca.cinvestav.mx, http://www.tue.nl

Abstract. This article extends ideas from crowd dynamics to a navigation controller for mobile robots. Each mobile robot is considered as an agent, associated to a comfort zone with a certain radius, which creates a repulsive force when this comfort zone is violated by its environment or by another agent, therefore avoiding collisions. Meanwhile, attractive forces drive the agents from their instantaneous position to a goal position. The resulting navigation controller is tested by simulations and experiments. It is found that simulations capture the global dynamic behavior that is shown in experiments, showing robustness of the proposed navigation controller.

Keywords: Crowd dynamics · Behavior · Mobile robot · Obstacle avoidance · Navigation

1 Introduction

When developing autonomous vehicles two major issues have to be considered, first position determination of the vehicle, second obstacle collision avoidance. For obstacle or collision avoidance several methods are proposed, some of them based on potential fields or geometric relations between the vehicle and obstacles. Potential fields are in general based on external sensors, or when the environment is structured and well known, then internal sensors might be used. A potential field is constructed around the obstacle [2], such that the distance to the obstacle determines a repulsive force. Constructing a potential field around an obstacle implies knowing the dimensions of the object and its position on beforehand.

The so called Geometric Obstacle Avoidance Control Method, GOACM, [3], is based on onboard sensors. When one of the sensors detects an obstacle within a safety bound the GOACM is triggered. The GOACM uses the distance and angle related to the sensor to determine a collision free way point, that is commonly located on a specified distance on a perpendicular line to the obstacle.

Y. Tan et al. (Eds.): ICSI 2016, Part II, LNCS 9713, pp. 553–560, 2016.
DOI: 10.1007/978-3-319-41009-8_60

When considering biological navigation systems, we can find schools of fish, herds of quadrupeds, flocks of flying birds, and human crowd motion. A way to model such systems is by considering self-driven particles, see [1]. Research on crowd dynamics helps predicting large crowd behavior at festivals or football matches as described in [4,5]. When information regarding the movement of the crowd is known, it can be used to describe natural phenomena like flocking behavior of birds, herds of animals or bacterial groups.

In this article the goal is to provide autonomy to a differential mobile robot, considered as an agent, in order to move from an initial to a goal position, evading dynamic and static obstacles, such as other agents and environment boundaries. Our controller is based on the crowd dynamics model introduced by Helbing [5], which describes the behavior of multiple agents with a generalized force model. Each agent has its own comfort zone, which when violated by another agent, wall or static obstacle, generates a repulsive force to drive the agent away from the obstacle. When the agent is free of collision, the dominant forces are attractive to a goal position, for that, straight line direction and distance are computed from the actual instantaneous position to the goal position.

In [5] the generalized force model is applied on the dynamics of a mass point, and the goal is to move in a prescribed direction at a desired velocity. Extending the ideas from Helbing [5] to the case of unicycle mobile robot is not immediate, since the last one is generally model at kinematic level and control by translational and angular velocities, while in [5] acceleration of the the mass point is considered. In this article, an extension is proposed resulting in a bio-inspired autonomous navigation controller, such that navigation performance resembles the behavior of an individual in a crowd and constraint environment, when moving to a desired position. The proposed navigation controller is tested by simulations and experiments by using several e-puck mobile robots.

2 Crowd Dynamics Model

The generalized force model as described by Helbing *et al.* [5] assumes the forces on each agent to consist of both socio-psychological and physical forces. By adding up these forces the total resulting force can be calculated as in (1), where it is considered a particle of mass m_i moving with a velocity vector $\mathbf{v}_i(t)$ and acceleration vector $\dot{\mathbf{v}}_i$, such that v_{x_i}, v_{y_i}, $\dot{v}_{x_i}$ and $\dot{v}_{y_i}$ correspond to the velocity and acceleration vector components in x and y direction respectively, $\mathbf{f}_{ij}$ represent the reaction forces among ith agent and other agents, and $\mathbf{f}_{iW}$ the reactions forces between ith agent and its environment or obstacles.

$$m_i\dot{\mathbf{v}}_i = m_i\frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)}\mathbf{f}_{ij} + \sum_{W}\mathbf{f}_{iW} \tag{1}$$

The first term in (1) causes each agent to regulate its velocity $\mathbf{v}_i(t)$ to a desired velocity of magnitude v_i^0, in a desired direction $\mathbf{e}_i^0$, with a characteristic time τ_i. All agents thus move towards their own specified desired direction which is why Helbing *et al.* [5] describes this term as socio-psychological.

The terms $\mathbf{f}_{ij}$ generate repulsive interaction forces between agents i and j.

$$\mathbf{f}_{ij} = \left[A_i e^{[(r_{ij}-d_{ij})/B_i]} + kg(r_{ij}-d_{ij})\right]\mathbf{n}_{ij} + \kappa g(r_{ij}-d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij} \tag{2}$$

The tendency for agents to stay away from each other is given by $A_i e^{[(r_{ij}-d_{ij})/B_i]}$, with constants A_i and B_i. The sum of the radii of two agents comfort zones r_i and r_j is r_{ij}. The distance between two agents is $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$, with $\mathbf{r}_i$ and $\mathbf{r}_j$ their positions. The normalized vector from agent j to agent i is $\mathbf{n}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/d_{ij}$. The terms $kg(r_{ij}-d_{ij})\mathbf{n}_{ij}$ and $\kappa g(r_{ij}-d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij}$ render normal and tangential forces respectively, that are present when agents get in each others comfort zone. The normal and tangential forces, push and turn the agents away from each other respectively. Function $g(x)$ is zero if $d_{ij} \geq r_{ij}$ this is when the comfort zones aren't violated; and is equal to its argument when $d_{ij} < r_{ij}$. The tangential vector is $\mathbf{t}_{ij} = (-n_{ij,y}, n_{ij,x})$, based on the normal vector $\mathbf{n}_{ij} = [n_{ij,x}\ n_{ij,y}]$. The tangential velocity difference can be expressed as $\Delta v_{ji}^t = (\mathbf{v}_j - \mathbf{v}_i)\cdot \mathbf{t}_{ij}$.

The terms $\mathbf{f}_{iW}$ are interaction forces between agent i and its environment or static obstacles, it depends on the approaching velocity of *i-th* agent.

$$\mathbf{f}_{iW} = \left[A_i e^{[(r_i-d_{iW})/B_i]} + kg(r_i - d_{iW})\right]\mathbf{n}_{iW} + \kappa g(r_i - d_{iW})(\mathbf{v}_i \cdot \mathbf{t}_{iW})\mathbf{t}_{iW} \tag{3}$$

The distance between agent i and the static obstacle W is d_{iW}, while $\mathbf{n}_{iW}$ is the normalized vector pointing from the obstacle W to agent i; $\mathbf{t}_{iW}$ is the tangential vector. In (2) and (3), k and κ are constant parameters that can be changed to influence the magnitude of the normal and tangential forces respectively.

2.1 Autonomous Navigation Controller Design

Each agent is represented by the kinematic continuous-time model of a unicycle mobile robot (4), with x-position, y-position and orientation angle θ, see Fig. 1. The inputs are the forward velocity and angular velocity (v, ω).

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i \\ \dot{y}_i &= v_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{4}$$

To extend the generalized force model (1), to the kinematic model (4), the variables v_{x_i} and v_{y_i} (components of vector $\mathbf{v}_i$), are transformed to v_i and ω_i.

Mobile Robot Velocity Inputs from Generalized Force Model. The kinematic model input v_i is the magnitude of the velocity vector $\mathbf{v}_i$, i.e.

$$v_i = \sqrt{(v_{x_i})^2 + (v_{y_i})^2} \tag{5}$$

Differentiating (4) with respect to time results in acceleration in x and y, and then it can be established that

$$\ddot{y}_i \dot{x}_i - \ddot{x}_i \dot{y}_i = v_i^2 \omega_i \tag{6}$$

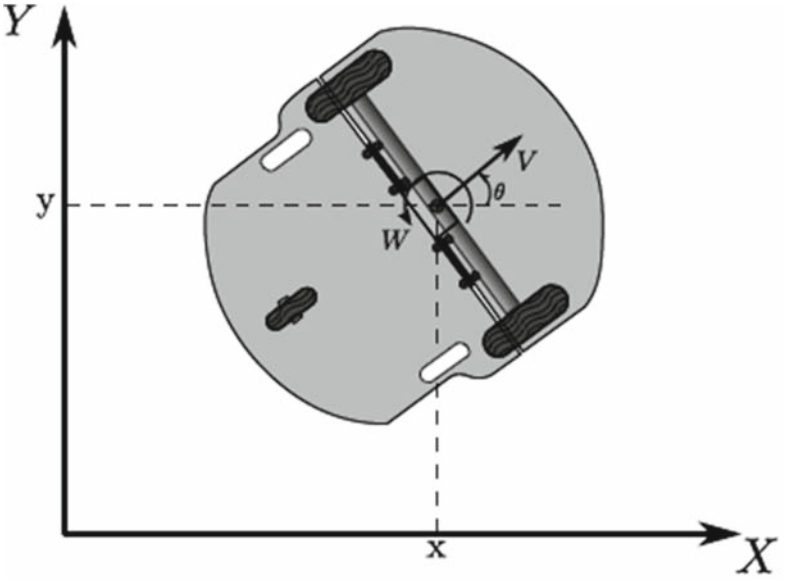

Fig. 1. Differential mobile robot.

From (1) and (4), we can set that $\dot{x}_i = v_{x_i}$, $\dot{y}_i = v_{y_i}$, $\ddot{x}_i = \dot{v}_{x_i}$ and $\ddot{y}_i = \dot{v}_{y_i}$, such that (5) and (6) yield the kinematic control input ω_i (7), where a positive small parameter $\epsilon \approx 0$ is introduced to avoid singularity when $v_i = 0$.

$$\omega_i = \frac{\dot{v}_{y_i} v_{x_i} - \dot{v}_{x_i} v_{y_i}}{\epsilon + \sqrt{(v_{x_i})^2 + (v_{y_i})^2}}, \tag{7}$$

Regulation to a Goal Position. Model (1) focuses on a particle moving along a desired velocity direction $\mathbf{e}_i^0$ with a desired magnitude v_i^0, as long as the *i-th* agent does not interact with other agents or obstacles. However, our goal is to move from actual (x_i, y_i) to a desired position (x_{d_i}, y_{d_i}). Thus, $\mathbf{e}_i^0$ is proposed as the difference between actual and desired position.

$$\mathbf{e}_i^0 = \begin{bmatrix} x_{d_i} \\ y_{d_i} \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{8}$$

Equation (8) provides the vector and the distance between actual and goal position for *i-th* agent, thus the agent will move faster as farther from the desired position, and slower as closer to it; being still when getting the desired position. This might result in unsatisfactory performance because of slow convergence, [6]. To render faster convergence, normalization of $\mathbf{e}_i^0$ can be considered. Thus the desired velocity magnitude v_i^0 would be imposed if the agent is free of interaction with other agents or obstacles, and when the desired position is achieved $\mathbf{e}_i^0 = 0$. In this article normalization of $\mathbf{e}_i^0$ is considered for the simulation and experimental results.

3 Simulation and Experimental Results Comparison

To investigate the performance of the proposed controller, (5, 7, 8), six e-puck robots are used, [7]. With a camera system the position and orientation of each robot are determined, in order to compute the control signal for each robot, which is sent by wireless communication. For comparison, a simulator is made in Matlab, that creates a movie showing the movement of each agent. The blue lines and shapes represent the environment which, in this case is represented by multiple walls and a column. The experimental initial positions and orientation angles of the e-pucks, are set as initial conditions in the simulator.

Simulations and experiments are done in two different environments. The first one has a column obstacle in the middle, Fig. 2. The second one has multiple walls with a bottleneck through which the robots pass, Fig. 3. In both simulations and experiments $A_i = 0$ and $B_i = 1$, since these parameters influence the tendency for the agents to stay away from each other and obstacles. Normally the exponential term in (2) and (3) would cause the robots to repulse from obstacles, even though their comfort zones are not violated. By setting $A_i = 0$ this effect is removed.

Column Obstacle Scenery. Comparison between experiment and simulation is shown in Fig. 2. Blue lines are experiment paths while red lines are simulation paths. In this experiment/simulation six e-pucks are initially positioned left of a column obstacle and the robots are trying to move in a straight line to their target position at the right. As soon as an obstacle comes within range of their comfort zone, it is expected that they start to avoid it. The same is expected if the e-pucks are too close with respect to each other. The experiment and simulation parameters are the same for all agents, and are listed at Table 1. Table 2 presents numerical results of the simulation (x_s, y_s) and experiment (x_e, y_e), for initial time $t = 0$ and final time t_f.

Bottleneck Scenery. Figure 3 shows a comparison between experiment and simulation, blue lines are experiment paths and red lines are simulation paths. Notice that in order for the e-pucks to reach their final position at the right, they must pass through the small bottleneck in the middle, causing possible collisions. The parameters for this experiment are the same for all agents and can be seen in Table 3. Meanwhile Table 4 presents the numerical results of the simulation (x_s, y_s) and experiment (x_e, y_e) for initial time $t = 0$ and final time t_f.

Table 1. Experiment and simulation parameters, column obstacle scenery

Parameter	A_i	B_i	k	κ	τ_i	v_i^0 [m/s]	r_i [m]
Value	0	1	2	1	0.5	0.05	0.1

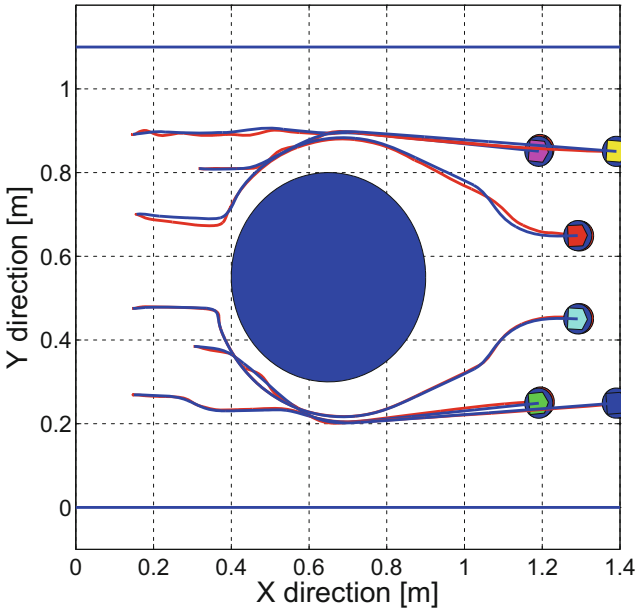

Fig. 2. Comparison between simulations and experiments, column obstacle scenery (Color figure online)

Table 2. Experiment and simulation results, column obstacle scenery

Robot	x_d	$x(0)$	$x_s(t_f)$	$x_e(t_f)$	y_d	$y(0)$	$y_s(t_f)$	$y_e(t_f)$
Green	1.2	0.1492	1.1912	1.1949	0.25	0.2689	0.2491	0.2518
Purple	1.4	0.3046	1.3903	1.4049	0.25	0.2689	0.2493	0.2489
Blue	1.3	0.1463	1.2903	1.2959	0.45	0.2689	0.4505	0.4510
Red	1.3	0.1567	1.2916	1.2961	0.65	0.2689	0.6494	0.6488
Yellow	1.4	0.3174	1.3908	1.4075	0.85	0.2689	0.8507	0.8498
Pink	1.2	0.1470	1.1911	1.1941	0.85	0.2689	0.8508	0.8551

Table 3. Experiment and simulation parameters, bottleneck scenery

Parameter	A_i	B_i	k	κ	τ_i	v_i^0 [m/s]	r_i [m]
Value	0	1	4	2	0.5	0.05	0.1

Table 4. Experiment and simulation results, bottleneck scenery

Robot	x_d	$x(0)$	$x_s(t_f)$	$x_e(t_f)$	y_d	$y(0)$	$y_s(t_f)$	$y_e(t_f)$
Green	1.4	0.2072	1.4002	1.3955	0.8	0.2555	0.7982	0.7936
Purple	1.4	0.1958	1.4008	1.4029	0.6	0.4727	0.6017	0.6082
Blue	1.4	0.1912	1.3987	1.3933	0.4	0.6830	0.3987	0.3786
Red	1.0	0.3761	0.9983	1.0028	0.2	0.7441	0.2001	0.1980
Yellow	1.0	0.3773	1.0014	0.9940	0.8	0.3736	0.8009	0.8015
Pink	1.4	0.1971	1.4015	1.4107	0.2	0.9125	0.2006	0.1912

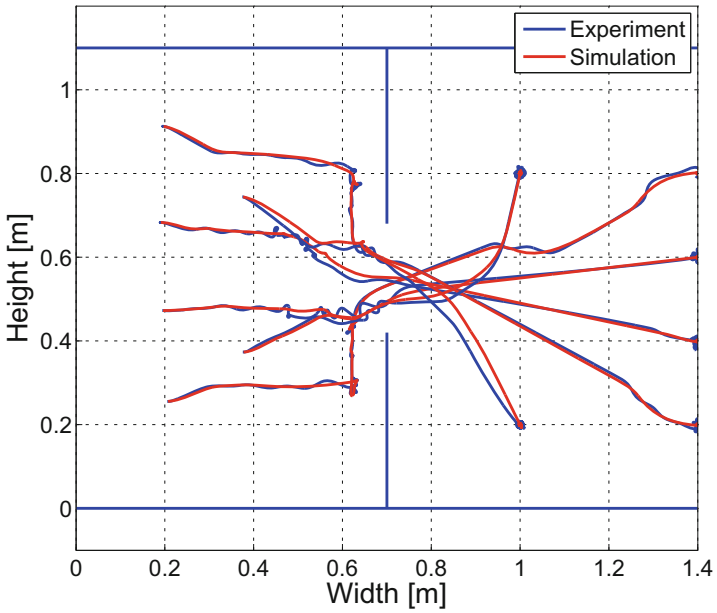

Fig. 3. Comparison between simulations and experiments, bottleneck scenery (Color figure online)

3.1 Results Discussion

Experiment and simulation produce a similar performance, see Figs. 2 and 3. Minimal differences between experiment and simulations, blue and red lines respectively, are noticed. All e-pucks move in a straight line to their target, until an obstacle or other agents get in their comfort zone, when they start turning away and going around, until they are free of collisions. One by one the robots are passing the column obstacle and the bottleneck in the middle until they reach their desired positions avoiding collisions.

Note that although the paths of the e-pucks in the simulation are not perfectly identical to the paths in the experiment, the simulation can roughly predict what the dynamics in the experiments are. The blue lines show that the e-pucks in the experiment have richer dynamics illustrated by the small bumps in their trajectories, Fig. 3. This happens when an e-puck has to make a fast turn, because violation of their comfort zone results in a repulsive force. Since the e-pucks are non-holonomic systems, the only way to react when they are pushed away is by making fast turns and going in the direction of the repulsive force.

Other differences between experiment and simulation are possibly due to the inaccurate model of the e-puck that is considered for the controller design, as well as shortcomings in the setup. The system that detects the position and angle of each e-puck has a poor performance, which could lead to errors. Another flaw in the setup is the dragging-behavior of the e-pucks, which are unbalanced and drag either with its front or back side of their frame over the ground. This results in an extra drag force which isn't accounted for. Another shortcoming is the sampling

rate which is limited to 30 Hz due to the frame rate of the camera. This could be a limiting factor when the e-pucks show very fast dynamic behavior.

4 Conclusions

A forced based autonomous navigation controller that steers robots from their actual to a goal position has been designed. The results of applying the proposed navigation controller happens to be a full analogon of what happens in crowd dynamics. Interesting enough- given the simple model, and all further shortcomings, the result features a great similarity of simulation and experiments and is thus very robust. It is shown that interaction forces between agents and environment can be used to react on real time, such that the agent can achieve a goal position by evading other agents and obstacles. Simulations and experiments show that violation of comfort zones leads to very fast turning behavior, instead of moving directly in the direction of the repulsive force. From comparison results, it can be concluded that although the simulations are not completely identical with the experiments, they do capture the global dynamic behavior. In conclusion this type of work is thus highly suitable for understanding and analyzing crowd dynamics, as well as for designing autonomous navigation controllers.

References

1. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. **75**(6), 1226–1229 (1995)
2. Kostic, D., Adinandra, S., van de Caarls, J., Wouw, N., Nijmeijer, H.: Collision-free tracking control of unicycle mobile robots. In: Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC), pp. 5667–5672 (2009)
3. Dai, Y., Lee, S.G.: Formation control of mobile robots with obstacle avoidance based on GOACM using onboard sensors. Int. J. Control Autom. Syst. **12**(5), 1077–1089 (2014)
4. Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-organized pedestrian crowd dynamics: experiments, simulations, and design solutions. Transp. Sci. **39**(1), 1–24 (2005). doi:10.1287/trsc.1040.0108
5. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature **407**, 487–490 (2000)
6. Xuesong, C., Yimin, Y., Shuting, C., Jianping, C.: Modeling and analysis of multi-agent coordination using nearest neighbor rules. In: 2009 International Asia Conference on Informatics in Control, Automation and Robotics (2009)
7. http://www.e-puck.org/