CLAWAR 2012 – Proceedings of the Fifteenth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Baltimore, MD, USA, 23 – 26 July 2012

LOCAL DISTRIBUTED CONTROL FOR MULTI-ROBOT NAVIGATION

A. TERRONES*, R. ACUÑA, N. CERTAD, L. FERMÍN-LEÓN and G. FERNÁNDEZ-LÓPEZ

Simón Bolívar University, Mechatronics Group, ELE-328
Sartenejas 1080-A Miranda, Venezuela
** E-mail: aterrones@usb.ve*

In this work a distributed navigation control strategy for multi-robot systems based only in the local information provided by the robots is presented. Each robot calculates the vector forces required for navigation towards a goal avoiding possible obstacles on the workspace and traveling in formation with other members of the swarm. A control architecture was implemented for the generation of the vector force used for navigation, based on three basic behavior agents responsible of calculating individual force components needed for navigation to the target, obstacle avoidance and swarm navigation. Also, an event driven control system is in charge of assigning a priority level to each force component generated by the agents of basic behaviors. Results obtained in simulations show the effectiveness of the control system in the navigation of a system of multiple robots under different obstacle configuration.

Keywords: Swarm, Distributed Systems, Artificial Potential Fields, Vector Force Fields, Event Driven Control

1. Introduction

The problem of coordinated navigation arises when the task is considered too complex to be efficiently executed by a single robot. Tasks such as reconnaissance, rescue and cleaning are complex enough to be performed by one robot in an acceptable time limit.[1,2] However, the presence of several platforms working together may optimize runtime work,[3] allows the use of simpler robots and introduces system redundancy.[2] Raynolds[4] developed an algorithm for the simulation of birds and fish flocks by using only three rules responsible of collision avoidance and velocity matching between agents and centralization of the group. Arkin *et. al.*[5] address the problem of flock navigation using a creative approach based on potential fields. Olfati-Saber[6,7] attacks the problem of multi-robot navigation using a vector field control for each robot. Estévez *et. al.*[8] used dynamically varying potential fields with insistence mechanisms for the coordination of a group of robots. In order to avoid being trapped in local minima, random components were added to the fields. Gayle *et. al.*[1] proposed the use of global information system and vector force field control to achieve local coordination of the robots in the system. McLurkin[9] develops a set of individual behaviors that serve as a basis for emergent behaviors.

The control scheme proposed in this work is decentralized, each robot computes its own control action based on data acquired from the environment. For this purpose, the agents are in charge of generating force vectors associated with different basic behaviors and sum them according to a priority plan based in several events. To solve the local minima problems, we propose an auxiliary agent whose function is grouping the obstacles in clusters based on their proximity.

2. Control Architecture

The proposed control scheme (Fig. 1) is based on three main agents responsible for calculating the control actions necessary to navigate towards the target, avoid obstacles and travel together.

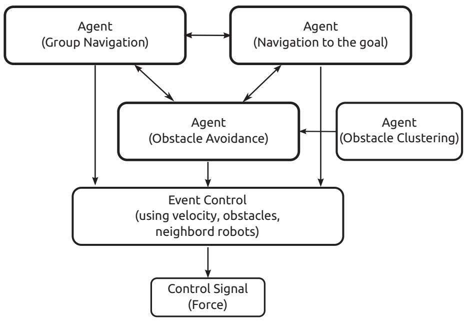


Fig. 1: General outline of a navigation controller based on agents and events.

Furthermore, there is one auxiliary agent in charge of the clustering of visible obstacles. Once calculated, the basic force vectors are added according to this equation

$$\mathbf{u}_i = \alpha_i \mathbf{u}_i^{target} + \beta_i \mathbf{u}_i^{group} + \gamma_i \mathbf{u}_i^{obstacle} - \delta_i \mathbf{u}_i^{damping} \tag{1}$$

were α_i, β_i, γ_i and δ_i are the coefficients of priority of each one of the components calculated in function of the states of each robot.

2.1. *Navigation to the goal*

The control function to the target is given by a simple potential well, which is described by the following equation

$$\mathbf{u}_i^{target} = max(\mathbf{q}_{target} - \mathbf{q}_i^{robot}, \mathbf{u}_{max}) \tag{2}$$

were $\mathbf{q}$ indicates a position vector. The Fig. 2 shows the graphical representation of the vector $\mathbf{u}_i^{target}$.

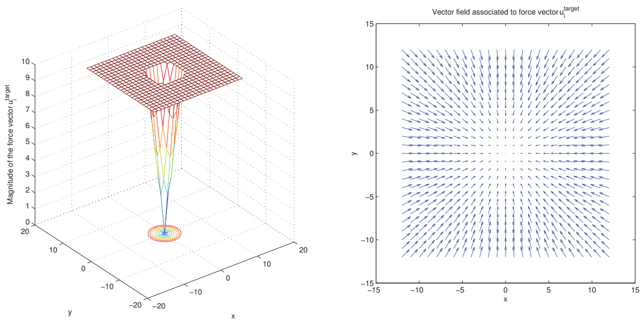


Fig. 2: Graphical representation of the force vector $\mathbf{u}_i^{target}$. The maximun value is set to 10.

2.2. *Group Navigation*

For the group navigation, a potential function with the following equation is used

$$\mathbf{u}_i^{group} = \sum_{j \in N_i} \left[\phi(\| \mathbf{q}_j^{robot} - \mathbf{q}_i^{robot} \| - d_{formation})\mathbf{n}_{ij} + c_1(\mathbf{p}_j^{robot} - \mathbf{p}_i^{robot}) \right] \tag{3}$$

based on the work of Olfati-Saber.[6,7] However, a different function is used for $\phi(n)$ in order to avoid oscillations around $x = 0$. The following equation define $\phi(n)$

$$\phi(n) = \frac{a+b}{2}\left(\sqrt{1+(n+c)^2} - \sqrt{1+c^2}\right) + \frac{a-b}{2}n \tag{4}$$

where a, b and c define the shape of the function.

The $\mathbf{q}$ and $\mathbf{p}$ vectors denotes the position and velocity of the robots respectively, and N_i is the robot neighbourhood consisting of those robots within the range of vision of robot i. The vector $\mathbf{n}_{ij}$ is a unit vector parallel to the vector of minimum distance between the robots j and i. The coefficient c_1 represents the K coefficient in a PID controller, which is responsible of velocity matching between robots. The parameter $d_{formation}$ is the desired distance between a robot and another. In Fig. 3 a graphical representation of the $\phi(n)$ function is showed.

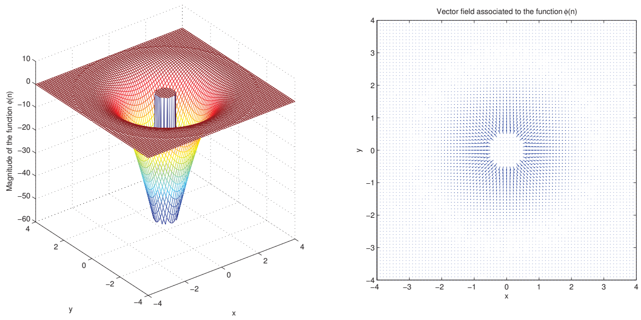


Fig. 3: Graphical representation of $\phi(\| \mathbf{q}_j^{robot} - \mathbf{q}_i^{robot} \| - d_{formation})$, with $a = 3$, $b = 45$, $c = 1.807$ and $d_{formation} = 4m$.

2.3. *Obstacle Avoidance*

For obstacle avoidance, the following control action is proposed

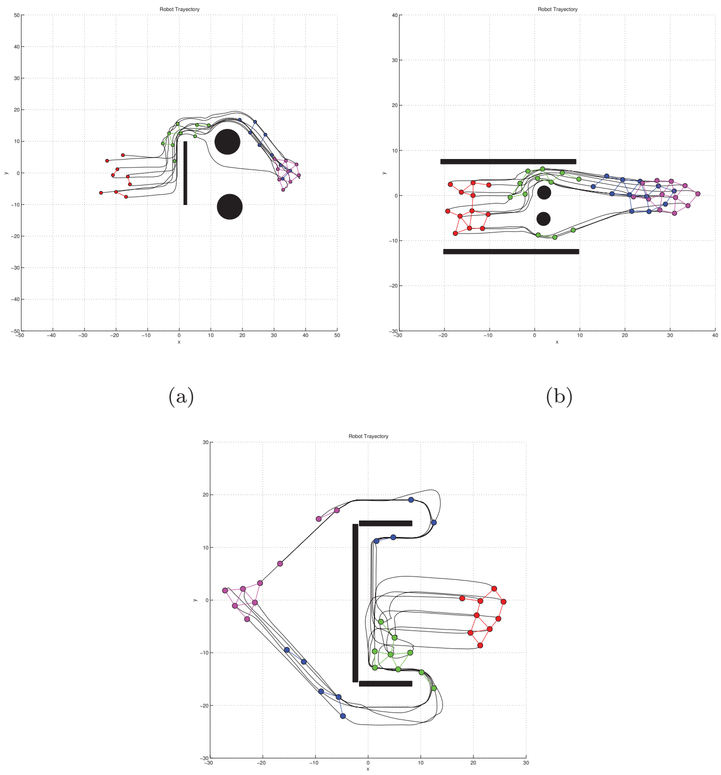
$$\mathbf{u}_i^{obstacle} = \mathbf{F}_g + \mathbf{F}_p + \mathbf{F}_{\mathbf{n}_{ij}}^{\perp} \tag{5}$$

were $\mathbf{F}_g$ corresponds to a gyroscopic force,[10,11] $\mathbf{F}_p$ is a potential field force defined by the function $\phi(n)$ defined previously and $\mathbf{F}_{\mathbf{n}_{ij}}^{\perp}$ is a force vector perpendicular to the distance vector between a robot and a obstacle.

The vector $\mathbf{F}_p$ is used to divert the robot from his original path, in order to avoid the incoming obstacle. $\mathbf{F}_p$ is responsible for keeping the robot at a certain distance from the obstacle. This approach allows the robot to be able to navigate through narrow areas without colliding with nearby obstacles. The vector $\mathbf{F}_{\mathbf{n}_{ij}}^{\perp}$ is used to allow the robot to surround the obstacles, which, in conjunction with the vector $\mathbf{F}_p$ and the obstacle clustering agent, allows the robot to escape from local minima.

2.4. *Auxiliary agents*

In addition to the basic behaviors agents discussed above, there are another agent whose function is to provide additional support for the obstacle avoidance behavior. This cluster agent is responsible for grouping the obstacles



(a) (b)

(c)

Fig. 4: Simulation results with different arrangements of obstacles: (a) Wall and 2 cylinders with target at (35, 0) and 9 robots, (b) Narrow corridor with target at (29, 0) and 12 robots, (a) C shaped obstacle with target at (-22, 0) and 9 robots. Different colors in each graph represents intermediate positions of each robot. Each colored line represents the line of sight between robots.

detected, so the robot can avoid being trapped in a local minima. To accomplish this, each distance vector to an obstacle detected is compared to the others to determine the angle formed between them. If this angle is less that a threshold, the vectors are then averaged to create a new virtual obstacle. This procedure is repeated until it is impossible to further reduce the number of obstacles.

2.5. *Priority System*

This system is responsible for modifying the values of the coefficients α_i, β_i, γ_i, δ_i as a function of several parameters such as: the presence of obstacles, number of robots belonging to N_i, robot velocity, relative location of neighboring robots and the target, so the robot can be adapted to the enviroment.

3. Results and Discussion

Simulation tests with a different number of robots and obstacles with different arrangements show the effectiveness of the implemented control architecture and its capability to achieve flock navigation under certain circumstances and avoid being trapped in local minima. The simulations were performed on a simulator developed by one of the authors, running on a PC with a Intel®Core$^{\text{TM}}$i7, with a time-step of $30ms$. All distances are in meters. The robots used in the simulations are of the holonomic type, with a maximun radius of vision of 4 meters, and a maximum speed of $2m/s$. The results of three performed simulations are showed in Fig. 4. The arrangement of obstacles for each simulation is such that it creates a potencial

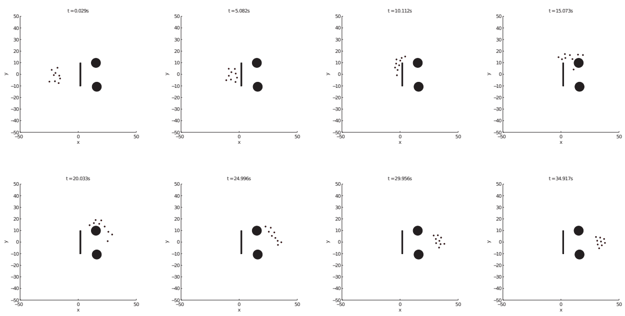


Fig. 5: Consecutive snapshots of the simulation results showed in Fig. 4a: Wall and 2 cylinders with target at (35, 0) and 9 robots.

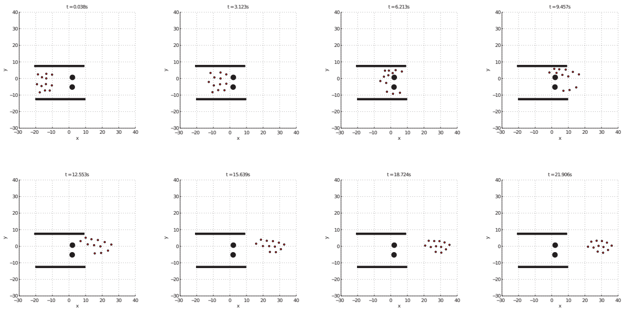


Fig. 6: Consecutive snapshots of the simulation results showed in Fig. 4b: Narrow corridor with target at (29, 0) and 12 robots

well around some obstacles. Traditional approaches to robot navigation and obstacle avoidance fail to prevent the robots of being trapped in a local minima. However, our approach succesfully demonstrates the ability to avoid being trapped in a potencial well, manifested in the fact that all robots, in each simulation, reach the target with no collision at all. The Fig. 4b shows the group's ability for navigation in narrow spaces while avoiding collision between them and the enviroment. Additionally, there is partial group fragmentation in each simulation, due to the fact that a total group cohesion is not absolutely requiered to accomplish their navigation objetive.

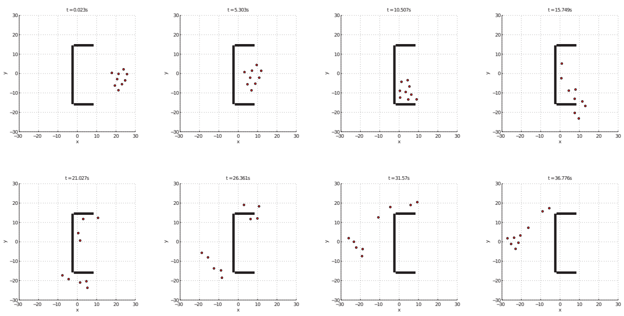


Fig. 7: Consecutive snapshots of the simulation results showed in Fig. 4c: C shaped obstacle with target at (-22, 0) and 9 robots

4. Conclusions

In this paper a distributed navigation control strategy for a multi-robot system was developed. This control system is based in three basics behaviors. In addition, a event driven control system was developed for managing execution priority levels of each basic behavior. To avoid potencial wells, an extra agent whose function is grouping obstacles in clusters was designed. Simulations performed for different workspaces shows the effectiveness of the algorithm allowing a group of robots to navigate in an unknown environment without colliding.

References

1. R. Gayle, W. Moss, M. Lin and D. Manocha, Multi-robot coordination using generalized social potential fields, *IEEE International Conference on Robotics and Automation* , 106 (2009).
2. W. Burgard, M. Moors, C. Stachniss and F. Schneider, Coordinated multi-robot exploration, *Robotics, IEEE Transactions on* **21**, 376 (2005).
3. N. Kwok, Q. Ha, V. Ngo and S. Hong, Particle Swarm Optimization-based coordination of a group of construction vehicles, in *Proc. Intl. Symposium on Automation & Robotics in Construction*, 2006.
4. C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH Computer Graphics* **21**, 25 (1987).
5. R. Arkin and T. Balch, Behavior-based formation control for multi-robot teams, *IEEE Transactions on Robotics* **XX**, 1 (1999).
6. R. Saber, W. Dunbar and R. Murray, Cooperative control of multi-vehicle systems using cost graphs and optimization, *Proceedings of the 2003 American Control Conference, 2003.* , 2217 (2003).
7. R. Olfati-Saber, Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory, *IEEE Transactions on Automatic Control* **51**, 401(March 2006).
8. P. Estévez, J. Cappelletto, F. Alvarez, R. Acuña, G. Fernández, J. C. Grieco and M. Armana, *Navegacion coordinada de un sistema multirrobot usando campos de velocidades dinamicos*, in *5º Workshop, Cooperación en Robótica*, 2009, pp. Vol. 1. pp. 79 – 103.
9. J. McLurkin, Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots, PhD thesis, (Massachusetts Institute of Technology, 2004).
10. D. Chang and J. Marsden, Gyroscopic forces and collision avoidance with convex obstacles, *New Trends in Nonlinear Dynamics and Control and their Applications* , 145 (2004).
11. S. Shadden, J. Marsden and R. Olfati-Saber, Collision avoidance for multiple agent systems, *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)* , 539 (2003).