Coordinated Object Manipulation by a Group of Robots

Roberto Hernandez[1], Rafael Castro[2], and Humberto Sossa[1]

[1] Department of Artificial Intelligence, CIC-IPN, Mexico D.F., Mexico
[2] Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico
rhar.ha@gmail.com, rcastro@cinvestav.mx, hsossa@cic.ipn.mx

Abstract. A control strategy for the control and coordination of two robots for the manipulation of objects is presented in this work. The ability to maintain a prescribed formation allows the robots to manipulate rigid objects, guaranteeing that the desired task is accomplished. The control goal is that the follower robots follow the leading robot with an unknown motion in the working area. For image analysis, an acquisition camera was mounted at the top of the working area. One of the primary advantages of this strategy is the facility to measure the position and orientation of all the robots, objects and obstacles. The vision system makes use of image analysis and pattern recognition techniques to allow the group of robots to accomplish their task. Preliminary experimental results performed with two Khepera II mobile robots show the effectiveness of the proposal.

Keywords: Mobile Robots, Artificial Vision, Pattern Recognition, Rigid Objects, Coordination of Robots.

1 Introduction

In order that a group of mobile robots follow a set of paths in working area to perform a task, it is necessary to know their trajectories by measuring their individual positions and orientations. For this, an image analysis and control system is needed.

The task of transporting objects by cooperative communities of robots includes, among several issues: avoiding collisions, calculating the trajectories of the robots, as well as the positions and orientations of the objects to be manipulated. Most robots must change not only their positions but also their directions to move the object through the work area, given that we have a fixed distance among the robots while transporting the object.

The usage of a community of robots allows efficiently manipulate objects better than with a single robot. Also the use of centralization information allows a total control of all the operations, processes and vision tasks.

In this work we propose a novel vision based-control law that, as we will see, allows manipulating of objects using a community of robots. Without a loss of generality, in this research we work with two robots only, one leader robot and one follower robot.

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 95-106

The rest of the paper is organized as follows. In section two we describe some of the most important works related with the present investigation. In section three we describe the proposed methodology, while in section four we present some initial experiments where our methodology is tested. Finally, in section five, we give some conclusions and directions for further research.

2 State of the Art

When we explore the literature we find either works related with the manipulation of objects by a single robot or works related with control of a formation of robots. Until now, to our knowledge, we had not found reported works discussing the problem of cooperative manipulation of objects by means of a formation control law. The only works that we have found in the literature that discuss the problem of cooperative object manipulation make use of heuristics [1] or centralized control [2].

In [1] the authors describe a methodology that allows a group of three robots to form a triangular formation. The leading robot has a set of marks in his posterior part that allow the follower robot to follow the leading robot. The follower robots have an acquisition camera mounted on them to perform the necessary image analysis to accomplish the task. Communication among the three is complicated, and most of the times delays among the robots is present, provoking that the triangular formation is lost.

In [2] a controlling computer makes use of a fixed camera mounted over the working area and odometry to determine the initial position and orientations of the robots and the objects to be manipulated. The computer then sends commands to the robots so they can follow their trajectories to reach a given object. In this case the control is an open loop. In the experiments presented by the authors, the robots reach their desired positions but only in an approximate way. The goal to manipulate a given is never accomplished as reported in the paper.

Other investigations have been also reported in the works [3] and [4], but they not deal with the accomplishment of the task of cooperative object manipulation, either because image analysis fails or because the desired robots are lost. A lot of work has to be performed in this direction.

In [5], robots calculate an actual relationship with a neighbor using only sensor readings. The robot communicates locally with discrepancies in its desired and actual relationships to neighboring cells. Correcting for these discrepancies produces robot movements that result the overall organization of the desired global structure. The achieved formations of the robots using sensors was extended by a robot that uses a video camera, but this also fails to make them work with real robots or manipulating objects.

In [6] behavior-producing modules are used: a robot is capable of following other while avoiding obstacles. Since no absolute referencing system is used, robots have to rotate themselves to detect the presence of the other robot nearby, the robots used the color camera according to the position of the colored-blob in the image to keep-formation leader. In this work, the qualifications obtained cannot be used to manipulate objects because they make triangular shapes or online.

2.1 Contributions

Given the published literature, this study provides some contributions that have not been done before and provided some modifications:

- Presents an alternative to manipulate objects using a visual control in a leader follower scheme.
- Is able to solve a small control errors caused by system dynamics.
- Generate of trajectories that can provide one of the minimal paths that may be able to avoid collisions.

3 Methodology

In this paper, the problem of cooperative object manipulation is divided into two stages, as follows. The first stage is composed of five parts:

1. Identification of the position and orientation of all items in the working area (robots, objects to be manipulated and barriers).
2. Determination of the shape of an object to be manipulated.
3. Determination of the number of robots needed to manipulate this object.
4. Determination of the path for each robot to reach their desired final positions.
5. Control of the robots to follow the computed trajectories to reach final manipulation positions.

Second stage is also composed of five parts, as follows:

1. Control of the robots to take the object.
2. Computation of the path of the leading robot to move in a cooperative way with the follower robots the object to a destination position.
3. Control the leader robot to follow the computed path.
4. Use of formation control low to move the whole set of robots to transport the object.
5. Put the object on the final position.

In the next sub-sections, we describe each if these steps in more detail.

3.1 Formation Control Model

In this work, a formation control based on the so called leader-follower scheme is considered [7]. Also, the controller for this formation is designed using passivity and sliding mode techniques in accordance to [8]. The formation control stage allows the movement of the follower robot to a given distance and angle with respect to the leader robot. The way in which these signals change is set in the stage of trajectory generation. The leader-follower formation considered is shown in Fig. 1, where $X - Y$ are the fixed coordinates and x$-y$ are the Cartesian coordinates fixed in the body of the leader robot. (X_L, Y_L) and (X_F, Y_F) denote the global positions of the leader and follower, respectively whit v_L and v_F are the linear velocities of the leader and follower; θ_L and θ_F are the respective angles of orientation; l is the relative distance between the leader and follower while φ is the orientation of the follower with respect to the leader.

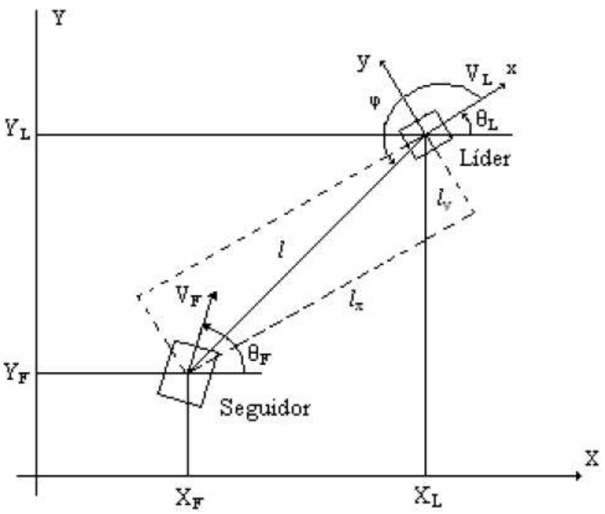

Fig. 1. Leader – follower formation.

The leader and follower kinematic models are then described by [9]

$$\dot{X}_L = V_L \cos\theta_L$$

$$\dot{Y}_L = V_L \sin\theta_L$$

$$\dot{\theta}_L = \omega_L \ . \qquad (1)$$

$$\dot{X}_F = V_F \cos\theta_F$$

$$\dot{Y}_F = V_F \sin\theta_F$$

$$\dot{\theta}_F = \omega_F \ . \qquad (2)$$

where ω_L and ω_F are the leader's and follower's angular velocities, respectively. Projecting the relative distance between the leader and follower along the x-y Cartesian coordinates leads to the expressions

$$l_x = -(X_L - X_F)\cos\theta_L - (Y_L - Y_F)\sin\theta_L$$

$$l_y = (X_L - X_F)\sin\theta_L - (Y_L - Y_F)\cos\theta_L \ . \qquad (3)$$

where l_x and l_y are the follower's relative positions along the x and y directions, respectively. In order to maintain a desired formation, it is needed that $l_x \to l_x^d$ and $l_y \to l_y^d$, where l_x^d and l_y^d are the desired relative position along the x and y directions, respectively.

Here, it is required that the desired relative distance between the leader and follower, l^d to be constant in order to be manipulate a class of rigid objects. On the other hand, to move the object requires that the desired angle φ^d to be time varying in accordance to a desired path. The following error signals are defined:

$$e_x = l_x^d - l_x,\; e_y = l_y^d - l_y,\; e_\theta = \theta_F - \theta_L \,. \tag{4}$$

Using the kinematic model (1), (2), the projections given by (3) together with some trigonometric identities and the so called *nonholonomic restrictions*, the dynamics of the error signals take the form

$$\begin{aligned} \dot{e}_x &= e_y \omega_L - v_F \cos e_\theta + f_1 + \Delta f_1 \\ \dot{e}_y &= -e_x \omega_L - v_F \sin e_\theta + f_2 + \Delta f_2 \\ \dot{e}_\theta &= \omega_F - \omega_L \,. \end{aligned} \tag{5}$$

where

$$\begin{aligned} f_1 &= -l_0 \dot{\varphi}^d \sin \varphi^d - \omega_L l_0 \sin \varphi^d + v_L \\ f_2 &= l_0 \dot{\varphi}^d \cos \varphi^d + \omega_L l_0 \cos \varphi^d \,. \end{aligned} \tag{6}$$

with $l^d = l_0 =$ constant Δf_1 and Δf_2 represent uncertain terms due to bad measurements or noise in the sensors and are bounded. It is important to note that f_1 and f_2 are known bounded functions since it is assumed that φ^d, ω_L and v_L are also smooth and bounded.

The control strategy for ω_F and v_F that would achieve the objective of manipulating some kind of rigid objects, even in the presence of the uncertain terms Δf_1 and Δf_2 is based on the results presented in [8] and [9]. This strategy consists of a control action when $\omega_L \neq 0$ (the leading robot turns around itself) and when $\omega_L = 0$ (the leader robot does not rotate around itself) and are briefly described in that follows.

3.1.1 Case when $\omega_L \neq 0$

In this case a control scheme based on backstepping, passivity equivalence and sliding mode techniques was used. This control is given by the controller

$$v_F = \alpha(e_y, y, t) - \omega_L s_1 - \omega - \gamma_2 sign(s_2)$$
$$\omega_F = \omega_L - k_3 e_\theta \quad (7)$$

where

$$\alpha(e_y, y, t) = \omega_L e_y + \dot{f}_1$$

$$-\frac{d}{dx}(\omega_L^{-1})[f_2 + k_1 e_y + \gamma_1 sign(s_1)] - (\omega_L^{-1})\dot{f}_2$$

$$-(\omega_L^{-1})k_1[-\omega_L y - k_1 e_y + \gamma_1 sign(s_1)]$$

$$-(\omega_L^{-1})\gamma_1 \frac{d}{dx} sign(s_1)[-\omega_L y - \gamma_1 sign(s_1)]$$

$$s_1 = e_y + k_1 \int_0^t e_y(\tau) d_\tau$$

$$s_2 = y - \int_0^t v(\tau) d_\tau$$

$$v = \omega_L s_1 - k_2 y$$

$$y = e_x - \alpha(e_y) \quad (8)$$

with

$$\alpha(e_y) = (\omega_L^{-1})[f_2 + k_1 e_y + \gamma_1 sign(s_1)] \quad (9)$$

γ_1, γ_2, k_1, k_2, and k_3 are positive real constants while the $sign(\cdot)$ function defined in the usual way and an approximation was used for this function so that the time derivative of $\frac{d}{dx} sign(s_1)$ in (8) could be evaluated.

3.1.2 Case when $\omega_L = 0$

The control action given by (7), (8) and (9) can only be used when $\omega_L \neq 0$. However, many trajectories imply the leader's orientation to be constant (for example, when the leader robot moves in a straight line). In this case the control strategy proposed in [7] was used and takes the form

$$v_F = c_1(\tanh e_x \cos e_\theta - \tanh e_x \sin e_\theta) + \xi$$

$$\omega_F = -c_2 e_\theta - c_3 sign(e_\theta) \tanh^2 e_\theta \quad \textbf{(10)}$$

where c_1, c_2 and c_3 are real positive constants with $c_3 > c_1$ and

$$\xi = \frac{f_1 \tanh e_x + f_2 \tanh e_y}{\delta + \pi(e_x, e_x, e_\theta)} \quad \textbf{(11)}$$

with

$$f_1 = -l_0 \phi^d \sin \varphi^d + v_L$$

$$f_2 = l_0 \phi^d \cos \varphi^d$$

$$\pi(e_x, e_x, e_\theta) = \tanh e_x \cos e_\theta + \tanh e_y \sin e_\theta$$

$$\delta = \begin{cases} 0, & \pi(e_x, e_x, e_\theta) \neq 0 \\ \delta_0, & \pi(e_x, e_x, e_\theta) = 0 \end{cases} \quad \textbf{(12)}$$

where δ_0 is an arbitrary small real positive constant.

It is important to notice that when integrating the control action (7), (8) and (9) together with the control action (10), (11) and (12) in the realization of a path for the manipulation of the rigid objects considered here a switching between the two control schemes should be considered.

3.2 Vision Control

The image analysis module makes use of a video camera placed at the top of the working area to identify the objects (robots, object to be manipulated and barriers) and to determine their positions and orientations. The camera is placed at a fixed height. We assume that objects are isolated from each other, and can appear in any stable position and orientation [10]. Classification of objects is obtained by means of a linear classifier (a distance classifier) which compares the describing vectors of the objects in the working area with the models stoked in the computer. The describing features used in this work are the compactness factor and first two Hu invariants to translations, rotations and scale changes [11]. First and second order moments were used to find the centers and orientations of the robots and objects in the working area. This information is passed to the generation and control modules to perform their tasks.

The vision module is responsible for monitoring the position of the robots that perform a task, using the camera fixed, specific data are collected robots and compared with those obtained from the trajectory generation process which represent the path to be followed by the robot at some point toward the state goal, whether it is the object or purpose final task in the event of a mistake this will be corrected by the control stage.

3.3 Trajectory Generation

In order to build a trajectory for the manipulation of the object, the working area is divided into cells in according to the size of the Khepera robot. Such a strategy leads to a path with minimum distances around each cell; this gives computational simplicity as proposed out in [12], the distances are then discretized using an appropriate metric based on the eight neighbors' connectivity of each cell and a label representative tag is assigned to each element of the working area.

Beginning at the point of destination, all the cells associated with the current value (in this case is two) plus one are update in the sense that it the cell does not have pre-assigned label, the updating is carried out with the remaining cells so that the source is reached. This guarantees that the trajectory does not pass thought a cell which is not achievable, such as obstacle.

To generate a path for the leader robot, one considers that moves from an origin cell to the next cell which has a lower numeric value; each cell value represents the distance that separates it from the point of destination. Fig. 2 shows the working area simulated in a computer describing the produce to generate a path from on origin cell. One may notice that the produce described generates paths of maximum longitude.

7	**18**	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Fig. 2. This shows the methodology of the path planning. The final position is the mayor number (*left–top*) and the robot position has number two (right-down), the obstacles are presents with number 1, while the other numbers could be found on the trajectory with minor path length.

The destination point of the first sub-stage is the position where the robots are to be located to manipulate the object. To calculate this, the relative position of the object seen from the video camera is used, drawing a circle of radius 2.3 inches from the center of each cylinder, establishing an ellipse whose focus are the angles between the axis and the centers of each cylinder. The crossing point between the focus and the ellipse is the final destination of each robot and its orientation is perpendicular to the axis.

The selection of the number of robots used in a task depends on the form that has the object to be manipulated, identifies the shape of the object through the chambers allowing video characteristic features of the object to gather and compare them with previously stored in a PC for this selection.

After knowing the number of robots involved in the task should create a path forward from the starting point (current position) to the final position using the method described above, this for each robot.

The manipulation of the object is coordinated by the PC which then sends the appropriate commands to the robots grasp and lift the object enough to be moved, this is achieved with a communication interface based on wireless sync, allowing parallel processing simultaneous sending commands [13].

3.4 Shape of Objects and Handling

There are two variants of objects:

1. The first consists of a cylinder that can be manipulated by a single robot.
2. The second is a rigid object having a structure similar to the table, which must be manipulated by two robots.

In the case of the second object is considerate of two robots for manipulation because the weight is greater than that which supports the robot with its gripper, assuming that the robot can load it as if it were a long tractor-trailer the subject difficult step towards your destination along the trajectory and potential obstacles that can be retrieved to find in their path.

The manipulation of objects is limited to pattern recognition supplied then through the stage of vision, to identify the type of object can determine how many robots are needed for the task of moving the object. But the crucial point of the manipulation is to detect the exact point at which to place each robot to manipulate the object, for this procedure are several factors involved such as:

1. The position and orientation of the object and the robots involved.
2. The final position where the object is placed.
3. The calculation of the final position of the robot path to the object 8que along this work will be appointed as the endpoint).

In the case of the cylinder, the end point is calculated according to the box with the least minimum distance but that is on the opposite side of the final position, while for the second object, it is necessary to make the calculation from end and considerate of the length of the gripper and the two robots must go hand in hand with the same orientation preference on the opposite side of the final position.

Once the robot is placed in the proper position to hold the object proceeds to send the appropriate commands for this not hold, the robot must close your tweezers and lift the object at a distance where the object does not hit the floor when is transported to its destination, nor should be raised that way too because it would cause friction and effort of the servo motors of the robots, 0.5 in was considered sufficient for handling.

4 Results

In this section we present some preliminary results that allow to verifying the effectiveness of the proposal. For this the tests were conducted in a virtual environment. Each stage was tested separately to ensure proper operation before being implemented in the real environment.

To model the physical properties of Khepera II robot and to implement the control low presented in section 3, simulation software was used [14]. In all cases, good results were obtained.

One of the found drawbacks was the initialization of the parameters of the robots. A correct parameter initialization is of vital importance. If they are not correctly initialized huge variations in the computed velocities of the wheels may appear. This might cause that these velocities cannot be implemented with real robots due to their physical constraints.

A solution found to this problem was to pose (position and orientation) the follower robot as close as possible to the virtual path it has to follow.

Path generation was tested on a virtual environment generated by a PC that models the real working area, by generating random conditions for the two robots and the objects to be manipulated. In this case, also good testing results were obtained.

Identification of the elements in the working area was successfully implemented by means of the same programming language used in [15]. The system was tested with 40 images of each item. A correct classification of all the objects was obtained in all cases by comparing the describing features of each element in the working area against their computed models. Also the positions and the orientations of the objects are computed.

In the Fig. 3 we show an example where the possibilities are calculated to move the robot to the final position, to find the minimum distance path that leads to the object enough to count the number of frames for which can happen, for it presents a comparative table showing the optimal path, The figure shows that for this example there are fifteen potential path that may follow, but in any case is the shortest distance and avoid all motor vehicle crash with an obstacle.

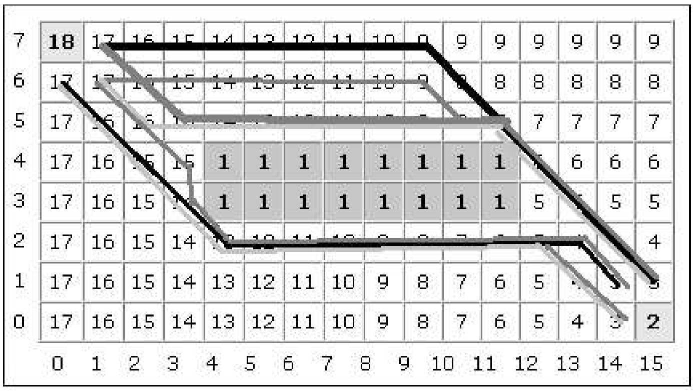

Fig. 3. The figure shows the possible paths solutions found after using the methods for finding the optimal path.

Because in all cases the same number of cells was obtained it is not necessary to present a comparative table of results, where travelling on paths yields the same time, because the cells are all of the same size.

5 Conclusion and Future Work

The paper is a compilation of different techniques and methodologies and others modified some proposals to present an alternative to the manipulation of objects using mobile robots.

The described path generation allows you found the path of least distance to the end point or end state, but not guaranteed to be the only solution will be sufficient for the work presented here. The control used allows follow leading a parallel robot which is suitable for handling objects as they can be transported in a way that is not a straight line.

This work shows the final results though, continues to change and some future changes are bluetooth incorporate a serial connection to replace the busy, you also have contemplated adding more robots for handling various objects that require manipulation of more than two robots.

Acknowledgements. We wish to thank *the Centro de Investigación en Computación* of the IPN by the support to accomplish this project. R. Hernández thanks CONACYT by the scholarship received to complete his doctoral studies. Authors thank the European Union, the European Commission and CONACYT for the economical support. This paper has been prepared under economical support of the European Commission (grant FONCICYT 93829). The content of this paper is an exclusive responsibility of the authors and it does not reflect the position of the European Union. Finally, authors thank the reviewers for their comments for the improvement of this paper.

References

1. Antonelli, G., Arrichiello, F., Chiaverini S., Rao, K.J.: Preliminary Experiments of Formation Control using the Null-Space-Based Behavioral Control. Indian Institute of Technology, Guwahati, (2006)
2. Borenstein, J., Everett, H. R., Feng, L., Wehe, D.: Mobile Robot Positioning, Sensors and Techniques. Journal of Robotic Systems. 14, pp.231-249 (1997)
3. Cheng, L., Wang, Y.: A communication-based multirobot rigid formation control system: design and analysis. Int. J. Modelling, Identification and Control, Vol. 1, No. 1, (2006)
4. Renaud, P.: Vision-based mobile robot formation control. (2004)
5. Mead, R., Weinberg, J. B., Croxell, J. R.: A Demonstration of a Robot Control Algorithm and Platform. Illinois, USA (2007)
6. Lemay, M., Michaud, F., Létourneau, D., Valin, J.M.: Autonomous Initialization of Robot Formations. Quebec, Canada.

7. Li, X., Cai, Z.: Backstepping Based Multiple Mobile Robots Formation Control. In: IEEE International Conference on Intelligent Robots and Systems. Edmonton, Alberta, Canada (2005)
8. Castro, R., Alvarez J., Martinez, J.: Robot Formation Control using Backstepping and Sliding Mode Techniques. Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico (2009)
9. Dixon, W.E., Dawson, D.M., Zergeroglu, E., Behal, A.: Nonlinear Control of Wheeled Mobile Robots. In: Thoma, M., Morari, M. (eds) Springer, USA (2001)
10. Sossa, J.H.: Feature Descriptors for Object Recognition. IPN, Mexico (2006)
11. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. In: IRE Transactions on information Theory, pp. 179-187 (1962)
12. Casarrubias, H.: Trajectory Generation for Mobile Robots Khepera II Using Machine Learning Techniques. Thesis, CIC-IPN, Mexico (2007)
13. Cheng, L., Waing, Y.: A Communication-Based Multirobot Rigid Formation Control System: Design and Analysis. Modeling Identification and Control, Vol. 1. N. 1 (2006)
14. MATLAB 2010a, http://www.mathworks.com
15. JDK 6 with Netbeans, http://java.sun.com