A Swarm Robotic Approach to Distributed Object Pushing Using Fuzzy Controllers

Aida Khozaee∗ Abdol Hossein Aminaiee† Ali Ghaffari∗

∗Faculty of Mechanical Engineering, K.N.Toosi University of Technology, Tehran, Iran
†School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

Abstract—This research is an attempt to design and develop behaviors for a group of robots to handle an object. A new approach to develop an object pushing system with a large group of mobile robots is introduced. The behaviors are devised in such a way that the robots would easily develop a swarm formation around the object and push it toward the desired goal. A rotating orbit of robots around the object is developed first, and the object is moved toward the goal as the orbit moves to the desired goal configuration.

Using a simple decision making, each robot can easily determine its proper action to push the object while remaining in the robot formation. Fuzzy controllers are used to develop behaviors in each robot. In this method, each robot performs individually in the system without a need to any information about other robots in the system. The main characteristics of the developed system are independency of the method to the object kinematics and its shape. Moreover, the method is not dependent on the number of robots in the system. Simulation results are given to support the proposed approach.

I. Introduction

Object manipulation is one of the appealing robotic topics which has attracted multi-agent and robotic communities. Many researches have been interested in pushing the object using a small group of agents to achieve the task [1], [2]. Most of these methods are dependent on the number of robots in the team. Besides, previous works have a complicated control algorithm and coordination method in the team. Furthermore, involving larger amount of robots to conquer the object manipulation task leads to a more fault tolerant system but with more complexity drawbacks in the cooperation and coordination methods.

Several techniques were introduced to reduce the complexity in such systems; Taking advantage of task assignment in [2] provides a fault-tolerant system with a simple algorithm where all the complexity were resolved by the use of a proper mechanics. In [3] a learning approach to object pushing was introduced to overcome the complexity in designing cooperation and coordination protocols between agents. A caging algorithm is proposed in [4] to entrap the object in a group of robots formation and move the object while the team of robots move toward the goal maintaining their formation.

Using a swarm of robots have been of great interest in robotics and artificial intelligence communities in recent years. Understanding nature such as social insects, birds and fish (Fig. 1) helps designing swarm systems in robotics and intelligent systems. Artificial intelligence utilizes swarm robotics to build systems consisting a large number of autonomous robots. Using a huge number of robots assists in achieving a specific task without a need to a centralized control or an explicit communication. Each unit simply detects the state of its environment and reacts accordingly with primitive actions supposedly known by the whole group [5]. Making use of simple and primitive actions in such systems directs to a simpler approach compared to multi-agent systems [6]. These characteristics make swarm robotic suitable for tasks requiring robustness, flexibility and scalability. In addition these systems can scale up and down in time, providing redundancy in performing the task [7]. Contributions of swarm robotics to object manipulation include [8], [9], [10]. In [8] a swarm approach to object manipulation is introduced where robots can pull a heavy object in cooperation. [10] introduced a swarm approach to object pushing. Object pushing by caging technique using a swarm of robots is also discussed in [4], [11] where in [11] robots are able to control position of the object without any controllability on the object orientation.

In this paper, a swarm approach to object pushing is introduced. Robots in this system do not acquire any communication with each other. The proposed method is independent from the number of robots available in the system. Simulation results show that the object can be moved on any desired trajectory without any requirement to object information such as its shape and its center of friction. A simple fuzzy inference system is used by each robot to provide the best action by each individual. Emergence of
**approach** and **surround** behaviors by the use of the **pushing** and **following** behaviors is another attribute of the proposed method.

The rest of this paper is organized as follows. In section II the scope of the paper, together with the problems which arise in the scope is discussed. Next, a model of the system is introduced in section III. The proposed method is presented in section IV followed by the simulation results in section VI. The paper is concluded in section VII.

### II. Problem Description

In this model a large group of robots are to push a two dimensional object on a frictional surface. Assigning a group of robots to push the object at all edges of the object enables the system to apply force and torque in any direction. This permits the object to be pushed on any desired trajectory. It is also desired that no dependency on any unit to be existed in the system. Independency of the method to the object shape and size is an attribute required to be realized.

A group of disc shaped nonholonomic mobile robots are exploited for the object pushing task. Using nonholonomic mobile robots makes it difficult to push the object while following the object movements in all directions on its path. Hence, a proper pushing and following behaviors have to be designed to cope with this problem. Moreover, as robots may push the object against each other, pushing behavior has to be designed in a way to prevent conflicts and possible lock in the system.

### III. System Model and Assumptions

It is assumed that the object is two dimensional with either concave or convex shape, which is pushed on a frictional surface by a group of robots. The object has \( M \) units of mass and \( I \) units of moments of inertial. By the assumption that there is no external vertical force on the object and a monotonous object contact with the surface, center of friction of the object is located on the center of gravity of the object (Fig. 2).

The amount of the frictional force between the object and the surface \( (f_f) \) is constant against the velocity of center of gravity of the object. The amount of the torque generated by the friction force is also constant with its direction against the direction of the angular velocity of the object.

Hence, kinematics equations of the object in existence of \( n \) number of different forces applied to the object is as follows:

\[
M\ddot{\mathbf{R}} = \sum_{i=1}^{n} \mathbf{F}_i - f_f \frac{V}{||V||} \tag{1}
\]

\[
I\ddot{\omega} = \sum_{i=1}^{n} \mathbf{r}_i \times \mathbf{F}_i - \tau_f \frac{\dot{\omega}}{||\dot{\omega}||} \tag{2}
\]

Fig. 3 shows a model of robots used in the simulations. Each robot has two parallel wheels with the same distance to the center of the robot. To eliminate the complexity of robot-object contact, it is assumed that the wheels of the robot reside inside the robot shape. Therefore equation 3 holds:

\[
\frac{w^2}{4} + r^2 < R^2 \tag{3}
\]

where \( R \) is the radius of the robot, \( r \) is the radius of the robot wheels, and \( w \) is the distance between the two wheels of the robot.

### IV. Proposed Method

Considering the nonholonomicity of robots, in order to enable the robots to push the object while following object movements in all directions, it is necessary that the robots have variable contact point with the object. Hence, robots would rub against the object edge to apply a force while changing its relative position to the object as in Fig. 4(a).

Rubbing behavior is principally similar to what was introduced in [12] where they used this behavior to reorient the object. In [12] the robot compliantly applies a sliding force on the object edge until the end of the edge is reached, then the robot would turn around to reacquire contact with the object and repeats the same actions until the goal is reached (Fig. 4). To simplify the rubbing behavior when the robot contacts the object edge, it is always set to push the object with a constant relative angle to the object edge (\( \frac{\pi}{5} \) in this paper).

We can consider a gyrate queue of robots that do the same behavior with enough number of robots to cover all the path that they move on (Fig. 5). This way, object movement would be speeded up due to the availability of at least one
robot, at any time, in front of the object edge to apply a force. Assigning a particular queue of robots to each edge of the object enables the system to apply a force and torque to the object in any direction (Fig. 5).

To increase the robot utilization in the system it is better to merge the groups of robots in each queue into one group by making all the robots move around the object (Fig. 6). Unlike previous explanations, each robot would move along the next edge if the end of current edge of the object is reached, instead of turning around and acquiring contact with the same edge of the object. Compared to the above paragraph, in this fashion all the robots maintain their contact with the object providing more utilization in the system and taking less amount of robots. To easily devise a formation in the team of robots, it is decided the robots to rub or follow against the object edge only in one direction. Hence, the robots would form an orbit around the object rotating in counter clockwise direction (Fig. 6).

Based on the relative goal configuration to the object edge, robot decides to push the object using its rubbing behavior or to follow the object edge (Fig. 7). In case rubbing the object edge applies a force or a torque against the goal configuration, robot will follow the object movements instead. Following the object will help the robot to remain in the robots formation rotating around the moving object. To follow the object, a robot controller is designed to make it move along the object edge with a constant distance to the object edge. To make enough room for the possible object movements, robot will move along the object edge with at least a predefined distance to the object edge.

Decision making procedure of an individual robot is independent from other robots information. This attribute not only simplifies the controller but also makes the method independent from the number of robots available in the system. Each individual performs its action only based on the relative goal information and its relative position. In other words, it acts as if no other robot is available in the system.

A. To Push or To Follow? That is the Question.

Fig. 9 shows the parameters which robots use to determine the proper action, whether to push or to follow. We define robot-object contact point as the closest point on the object edge to the robot, regardless of having a physical contact between the robot and the edge. In Fig. 9 contact points for Robot I and Robot II are shown by $P_1$ and $P_2$ respectively.

To determine whether pushing the object may result in a proper object movement toward the goal, the relative position of contact point to goal is used. Robot will push the object only if its contact point is placed between the robot and the extended corresponding edge of the goal. For instance, as $P_2$ is between the Robot II and the line $l$, Robot II decides to push the object while Robot I will only follow the object as $P_1$ is placed in the opposite side of line $l$. Fig. 8 shows how the two basic behaviors relate to each other in a subsumption architecture.

In the instance shown in Fig. 9, robots that move along $AD$ will follow the object edge while their contact position is placed on the line $DG$. They push the object only when their contact position is on $AG$. Accordingly, robots on the edge $AB$ are pushing the object. Robots painted in gray in
Fig. 9. A visualisation of the decision making algorithm and its corresponding parameters. The robots in gray are pushing while the robots in white are following the object.

Fig. 9 have selected their Pushing behavior while others are following the object.

Using this scheme, the groups of robots on any edge of the object tend to align the edge to the corresponding edge on the goal. It is obvious if all the edges of the object get aligned with their corresponding edges on the goal, the object will be placed on the desired goal position. Therefore, as the decision mechanism does not have any consideration to the object friction center and center of gravity of the object, it makes the algorithm independent from the object model and its shape.

V. CONTROLLER DESIGN

As robots perform in a continuous state and action space, fuzzy controllers are chosen to determine robots actions. Each robot performs individually in the system. In other words, the information about other robots is not included in the state space of each individual robot. Two separate fuzzy controllers are designed for the two main robot behaviours: namely, pushing and following.

The state space for the pushing and following actions are relative robot angle to the object edge ($\alpha$), robot distance to the edge ($d$), and relative robot position to the left point of the object edge ($l$) as in equation 4. Figure 11 shows a demonstration of these parameters.

$$S = (\alpha, d, l)$$

To make the fuzzy controller independent from the length of the object, the value of parameter $l$ is mainly calculated proportional to the edge length. Hence, the value of $l$ is calculated as in equation 5. The fuzzy sets used in the fuzzy controller are shown in Fig. 10.

$$l = \frac{|AP|}{|AB|}$$

Additionally, the action space is selected to be the robot’s left and right wheel velocities ($v_L$ and $v_R$). The nine possible pairs as the consequent of fuzzy rules in the fuzzy controllers are shown in Fig. 12. Each cell in the figure represents the possible consequent of a rule in the fuzzy controller.

VI. SIMULATION RESULTS

A couple of simulations are conducted to verify the proposed method. The object model used in the simulations is a two dimensional object of $1.7m \times 0.8m$ size and 5 units of mass and 0.5 units of moment of inertia (Fig. 13). The object coordinate frame is placed at $O$. The object reference point is also located at point $O$. The static and dynamic friction coefficients between the object and ground are $\mu_s = 0.8$
and $\mu_t = 0.3$ respectively. To model the friction between the robot and the object edge we have used a spring and damper mechanism with damping factor and spring coefficient as $B = 1$ and $K = 1k$ respectively. We utilized 20 robots in the simulations with $R = 15cm$ radius, in which the distance between the wheels are $w = 23cm$.

**A. Aligning a single object edge to the goal**

To proof that system can align a single arbitrary edge of the object to the corresponding edge on the goal, a number of simulations were performed. In this simulation we temporarily modified the algorithm to only follow the object edges except one edge. In other words, the robots just try to align a particular edge of the object to the goal. In simulation result shown in Fig. 14 robots could align the bottom edge of the object to the goal. Due to the undesirable errors in the system, position error and orientation error are not completely compensated. This is because robots do not push the object on the other edges. Hence, this disables the system to negate the extra object movements if the object passes by the goal edge.

**B. Pushing the object toward the goal**

In the simulation shown in Fig. 15 all edges of the object are set to be pushed if necessary according to the proposed method. The figure illustrates the track of the object as a result of commanding the group of robots to move the object from point $(1, -1)$ to $(-1, 2)$. The friction center is located on point $(0.85, 0.4)$ in object coordinate frame.

**C. Pushing an object with an arbitrary center of friction**

To test the independency of the method to the location of friction center, the simulation result presented in section VI-B is accomplished with a modification to the position of the object friction center. In this simulation, the friction center is placed at $(1.3, 0.2)$ in object coordinate frame.

Comparing results with previous simulation, increasing in the orientation error occurs at first. This is because of placement of the friction center close to the corner of the object. Hence, any force applied to the object may cause a great value of torque on the object which may lead to undesirable object rotations. But it can be seen that the system can compensate this undesirable object movements, leading the object toward the goal providing sufficient accuracy.

**D. Rotating the object around an arbitrary point**

To test the drivability of the system, the object is rotated around an arbitrary point. Fig. 17(a) shows the track of the object being rotated around the point $(0, 0)$ in the object coordinate frame. Fig. 17(b) illustrates the displacement of the desired center of rotation; point $(0, 0)$. It can be seen that robots could rotate the object around the center of rotation with quite small displacements.
E. Pushing an object of an arbitrary shape

A couple of simulations were performed to test the validity of the proposed method with objects of an arbitrary shape. Fig. 18 shows an example with an object of concave shape, and its track while being pushed from (−1, −1) to (0, 3). The results show the independency of the method to the object shape without any modifications to the controller, and parameters in the system. In this simulation the number of robots is set to 25.

VII. Conclusion

A new approach to cooperative transport of a polygonal object by means of a large group of mobile robots is introduced. Based on the proposed idea, robots form a rotating orbit around the object and move the object as the orbit moves toward the goal. In this method, each robot performs individually in the system without a need to the information from the other robots in the system.

Based on this idea, using a simple decision making, each robot could easily determine its proper action to push the object while remaining in the robot formation. Fuzzy controllers were used to develop each behavior in each robot. Two major behaviors of the robots are described. They have been designed in such a way that enable the robot to remain in the robot formation during object manipulation.

The main characteristics of the developed system were independence of the method to the object kinematics and its shape. Moreover, the method was not dependent on the number of robots available in the system. Simulation results were given to support the proposed approach.

Future works to our system include providing experimental results of the system performance. Additionally, we would like to mathematically analyze and proof the method. Modifying the pushing behavior to push the object with variable relative angle and compare the performance is another work in the future.

References