Enhancing the Awareness of Decentralized Cooperative Mobile Robots through Active Perceptual Anchoring

Sherwin A. Guirnaldo, Keigo Watanabe*, and Kiyotaka Izumi

Abstract: In this paper, we describe a system for controlling the perceptual processes of two cooperative mobile robots that addresses the issue of enhancing perceptual awareness. We define awareness here as knowing the location of other robots in the environment. The proposed system benefits from a formalism called perceptual anchoring. Here, perceptual anchoring enhances the awareness of the system by employing an anchor-based active gaze control strategy or active perceptual anchoring to control the perceptual effort according to what is important at a given time. By anchoring we extend the notion of awareness as knowing what the symbols in the control module represent to by connecting them to the objects or features in the environment. We demonstrate the present system through a simulation of two nonholonomic mobile robots performing a cooperative transportation by carrying a cargo to a target location where there are two other robots moving about. The system is able to efficiently focus the perceptual effort and thus able to safely carry the cargo to the target position.

Keywords: Awareness, decentralized control, mobile cooperative robots, perception.

1. INTRODUCTION

The success of any autonomous robotic task relies on the capability of the robot's perceptual system. In designing a robot's perceptual system (e.g., the choice of sensors), the designer is often influenced by the nature of the task at hand and often customizes the robot's design according to that task. However, despite careful customization, sensor inherent limitations remain an issue that must be addressed in designing autonomous robots. For instance, the commonly encountered problems of a typical camera include limited field of view and range, and occlusion, in which a vision system cannot see through walls or through the body of other robots and objects. In this sense, tracking all significant details in the environment is inconsequential if we would need controllers or behaviors to include them in decision making. A wider range of information will result in more reliable decision making.

Consider the situation depicted in Fig. 1, which shows two robots cooperatively transporting an object to a certain destination. Each of the robots is equipped with a panned camera system. To implement this system, it is necessary to identify the problems that must be addressed at the software level, e.g., recognition; each robot must be able to recognize landmarks to estimate its position and to aid its navigation system, and must be able to recognize other moving robots or objects and know their position in order to avoid collision. However, each of the robots can only view a fraction of the environment at any given time. This is dictated by the limitation of field of view and range of the sensor. In conventional

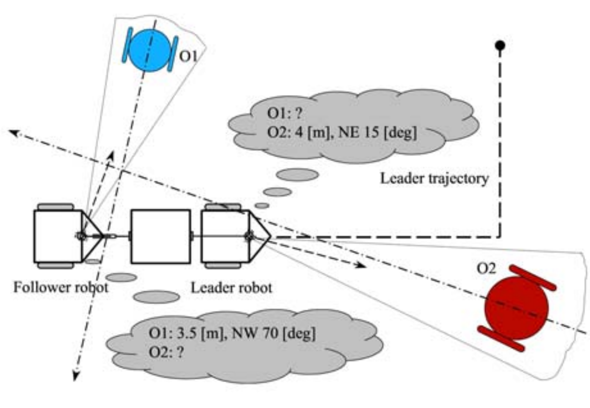


Fig. 1. Cooperation in a dynamic environment.

Manuscript received May 7, 2004; revised September 30, 2004; accepted October 6, 2004. Recommended by Editorial Board member Jae-Bok Song under the direction of Editor Keum-Shik Hong.

Sherwin A. Guirnaldo is with the Chair of Production and Control Technology, Department of Engineering Systems and Technology, Graduate School of Science and Engineering, Saga University, Saga 840-8502, Japan. His permanent address is the Department of Mechanical Engineering, College of Engineering, Mindanao State University, Marawi City, Philippines (e-mail: sheronell@yahoo.com).

Keigo Watanabe and Kiyotaka Izumi are with the Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, Saga 840-8502, Japan (e-mails: {watanabe, izumi}@me.saga-u.ac.jp).

* Corresponding author.

approaches, if the sensor of one robot is focused on one particular object the robot will then lose its awareness of the other objects in the environment. Therefore, each of the cooperative robots should have a facility that allows them to be conscious of the important details of the environment despite the sensors limitations. This facility should come in addition to the action processes that are responsible for producing control commands that directly guide each robot to achieve its goals. In effect, the designer must take into consideration that there will be two processes in each robot (for decentralized cases): one is the action process and the other is the perceptual process, which is responsible for prompting awareness and control to the sensors.

This paper focuses on developing a formalized approach to awareness, particularly for decentralized cooperative mobile robots. Specifically, the design of the architecture used for awareness was highly influenced by the need for separating the action processes and perceptual processes while maintaining some level of interaction between the two. This requirement is essential in view of the fact that the entire system is designed for performing various tasks in addition to monitoring and sensing.

2. COOPERATIVE ROBOTS AND SENSING

Recently, the interest in cooperative robotic systems has grown significantly (e.g., [1-5]). The primary reason for this increasing interest is the recognition of the large number of application domains in which cooperative robotic systems are applicable, including the following:

1. Military applications such as surveillance, reconnaissance, and de-mining.
2. Industrial applications such as cleaning, earth moving, and transportation of large objects.
3. Underwater and space exploration applications, such as pollution monitoring, rock gathering and search for water on other planets.

Advantages that can be achieved by using cooperative teams of robots include increased robustness through redundancy, decreased mission completion time through parallelism, and a potential to reduce the individual robot complexity through heterogeneous robot teams.

Many studies have focused on the development of cooperative control strategies that enable a group of robots to work together in many applications. Most of these studies were focused on designing cooperative control strategies. In general, control strategies are classified into centralized, distributed, and fully decentralized approaches [4]. The centralized approach is characterized by centralized processing of data, in which the processing location could be in one of the team members or in an external machine. All decision-makings are processed in that location and commands are sent to each team member from the processing unit. In the distributed approach, local data processing is performed at each individual robot but then information is sent to only one robot that will take the decisions. In a fully decentralized approach, each robot takes its own decision based on its own processing capabilities to process its own data and data shared by other teammates.

Although sensing is one of the fundamental issues of robotic systems, very little of the research for cooperative mobile robotics was aimed to address the issue of perception and efficient sensing. Parker [6] proposed to use a team of cooperative robots for the observation of multiple targets. The focus in [6] was to develop a distributed control strategy that allows the robot team to maximize the collective time during which each target is being observed by at least one robot team member in an area of interest. The key issues in this particular application are that of sensor placement - determining where each sensor should be located to maintain the targets in view. In addition to that, sensors have limited range and therefore the use of multiple sensors dynamically moving over time is required.

Specifically associated with cooperative mobile robotics is the need for each robot to take the others into account. Parker [7] also investigated the issue of robot awareness of team member actions and its effect on cooperative team performance by examining the result of a series of experiments on teams of mobile robots performing a puck moving mission. The result of Parker's study [7] indicates that the impact of action awareness on cooperative performance is a function not only on team size and the metric of evaluation, but also on the degree to which the effects of actions can be sensed through the world, the relative amount of work that is available per robot, and the cost of replicated actions. Moreover, Touzet [8], who defined robot awareness (for cooperative mobile robotics) as the perception of the locations and actions of other robots, proceeded to define four levels of awareness and studied their effect on cooperative mobile robot learning. Each level of awareness differs in regards to the amount of additional information. Touzet's results [8] indicated that awareness produced superior cooperation in a multi-robot observation of multiple moving targets.

Despite the convincing results shown above, none of them explicitly tackle the case of awareness for a cooperative mobile robot as a problem of controlling the perceptual effort. It appears that their common approach is to mold the cooperative behavior entirely to enable better awareness, thus making the cooperative behavior purely customized for enhancing

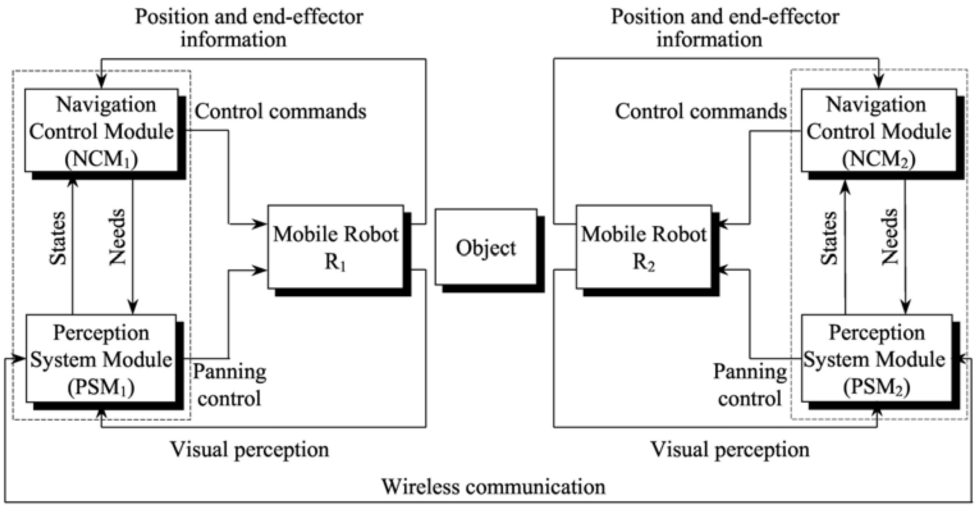


Fig. 2. Decentralized control strategy.

awareness. As stated above, the range of cooperative robot applications is beyond constructing simple observation or surveillance systems. In some applications, such as decentralized cooperative mobile robots that cooperatively transport a cargo according to Yang et al. [9,10], there is a need to separate the process of controlling the perceptual effort from some action processes of the cooperative behavior, while maintaining some form of interaction between the perceptual and action processes so as to allow the perceptual processes to efficiently conform their objectives for the needs of the cooperative behavior. Here, the present awareness is defined as knowing the position of other robots in the environment. The notion of awareness is then extended to knowing what each of the symbols (i.e., symbols in a controller) means or refers to — that is anchoring symbols to perceptual data that correspond to the actual objects or features in the environment. This new notion of awareness allows each robot to remember the position of other robots in the environment and does not just rely on fresh inputs from sensors or information from other robots passed through a communication channel.

For the needs of separating the perceptual processes from the action processes, the author developed a decentralized system for cooperative transportation of a cargo using two mobile robots. Each agent controller is composed of two modules, namely the navigation control module (NCM) and the perception system module (PSM). The PSM employs an active perceptual anchoring (APA) strategy [11]. The goal of employing an APA strategy is to enhance the perceptual awareness of the system by actively controlling the perceptual effort of a robot sensor. The APA is realized by a finite state machine that actively controls the focus of attention with the help of anchoring. Conversely, anchoring is the process of creating and maintaining the correspondence between

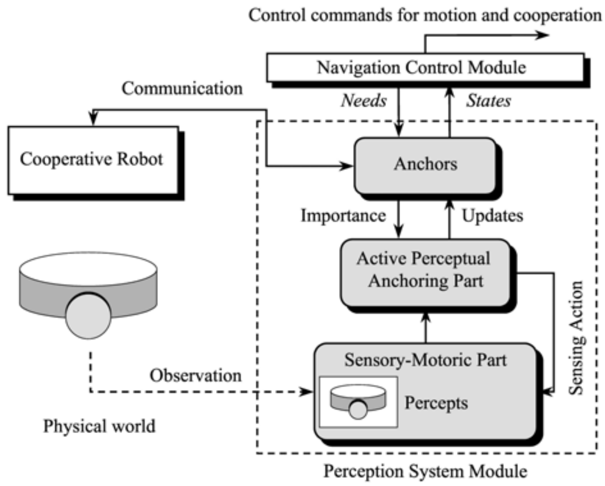


Fig. 3. Details of the perception system module (PSM).

symbols and sensor data that refer to equivalent physical objects [12-14]. Anchors will be used here as internal models of the relevant features and objects in the environment.

2.1. A decentralized system for cooperative robots

Fig. 2 shows a decentralized system for two cooperative mobile robots, R_1 and R_2. Two major components are visible in the figure: the NCM and the PSM. The NCM is in charge of generating control commands related to navigation and for cooperation with its partner robot. On the other hand, the PSM is in charge of controlling the perceptual effort of the system and consequently responsible for providing the awareness capability to each of the cooperative agents. The PSM will generate control commands to change or track the current focus of attention.

A more detailed look at the PSM is shown in Fig. 3. The PSM includes an APA element and anchors.

Anchors are simply data structures so that each anchor can contain several types of information pertaining to a particular object (or feature) in the environment. The NCM is supplied with information from the anchors by grounding each symbol with its corresponding objects (or features) in the environment through the anchors. The NCM has the ability to inform the PSM of the important objects (or features) at the current time by passing the *needed* measures of each symbol to the PSM. And in response, the PSM will ensure that those symbol-object connections are maintained by keeping the information in the anchors updated. The information in the anchors should be an accurate estimate of the actual state of those objects (or features) that are relevant to the NCM.

To update the anchors, the APA part will read each of the *importance* values of each anchor and use them to decide which, among the objects (or features) in the environment needs to be anchored. Consequently, the APA part will generate the necessary control commands to the actuator of the sensor to physically point the sensor along the expected direction of the most important object (or feature) in the environment. While the sensor of the agent continually sends percepts to the APA part of the PSM, if it detects one of those objects (or features) that are valuable to the agent mission, the APA part will measure its properties (e.g., distance) and send updates to the corresponding anchor.

Furthermore, the author included a wireless communication connecting the PSM of the two cooperative robots to allow the two cooperative agents to share the information that they are aware of. This kind of feature is somewhat human like, for instance, when two or more persons cooperatively transport a huge and heavy object, each person will usually share information regarding position of obstacles or position of the cargo that is otherwise not perceivable by other persons due to occlusion.

2.2. Active perceptual anchoring

The concept of APA yields the effect of combining together two popular approaches to perception control. One is the approach of packing together the perceptual and action processes into one module (or behavior) and another is the approach of using information about the current task to perform an active control of the agent's sensor [15-19]. The purpose of the former is to focus the perceptual effort exclusively on those features in the environment that are relevant to the current task. The latter is to actively control the agent's sensor that will allow the agent to search for features in the environment. Such an active control means selecting a specific algorithm or physically pointing a sensor in a specific direction; the concept of the active control was initially presented in [20] and presently it remains as one of the active research areas in computer vision [21-23].

The main advantage of employing APA and anchors in the PSM is that the APA part can use the information from the anchors to choose which among the objects (or features) in the environment, will be the focus of attention and can narrow down the search process. Each anchor can contain several types of information that best describe the state of the object that it represents. Conforming to our definition of awareness as knowing the position of objects in the environment, an anchor will contain data such as the relative orientation and distance from an observing robot to the object. As in [11], each anchor will also contain an *anchored* value on [0, 1] scale, which measures how recently the anchor was actually anchored (i.e., updated) to the real object in the environment. Moreover, each anchor will also contain an *importance* value that measures how important an anchor is to the PSM. For instance if the *importance* values in the anchors indicate that a certain object needs to be monitored at the present time, the APA part can simply use the estimated values of properties (e.g., relative position and distance) stored in the corresponding anchor to approximate the current location of that object in the environment, making the perceptual effort of searching and tracking the object more efficient.

Moreover, the PSM will receive a *needed* measure for each symbol from the NCM. The needed measure is a gauge that tells which of the objects is vital to the current state of the NCM; generally the higher the needed value the more valuable it is for the current state of the NCM. By allowing the NCM to pass the needed measure to the PSM and by making the important measure dependent on the needed measure, the perceptual effort generated by the PSM will be made relevant to the current state of the NCM. The importance measure is computed based on the needed measure and the anchored measure.

In the preceding section the author introduced the cooperative mobile robot platform, in which the author considered two mobile robots that cooperatively transport a cargo by carrying it to a desired destination (Fig. 1 will give a good illustration of the scenario). Moreover, the working environment is cluttered with two other moving robots. Therefore, each cooperative agent is required to have the ability to avoid collisions with other robots working within the same environment. Here, if the NCM of a cooperative agent is not in obstacle avoidance mode, the PSM will select the anchor with the highest *importance* value to be the focus of attention. For a given anchor a in S, where S is a set of anchors, and representing the important measure of a as

$$important(a) = needed(a)[1 - anchored(a)], \quad (1)$$

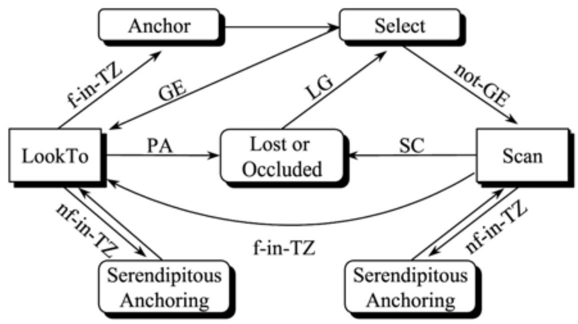


Fig. 4. Active perceptual anchoring system.

where $needed(a)$ is the needed measure of a and $anchored(a)$ is the anchored value of a.

In contrast, if the NCM is in obstacle avoidance mode, the PSM will only use the *needed* measure from the NCM to select the next focus of attention. The NCM must be supplied with fresh and accurate information relating to the most threatening object in the environment. In our simulation study, the *needed* value for the non threatening object is set to 0.5 and 1 if it is very close. Moreover, the *anchored* value for the newly updated anchor is set to 1; otherwise it is reduced by 1 percent at every time step.

Shown in Fig. 4 is a finite state machine that represents the processes of the APA part of the PSM. The finite state machine generates a sensor control command that can actively point the panned camera to the (new) fixation and will update the information in the anchors with percepts from the sensor. The details of each process are given below.

1. Select: choose an anchor x to become the new focus of attention of the panned camera. Set the fixation to the expected relative orientation of x. If the *anchored* level exceeds above from a given threshold, exit via GE (good estimate); otherwise, exit via the non-GE transition.
2. Scan: perform a visual scan to explore the portion of the space where x could be located. In our simulation, exploration is conducted by augmenting a search factor to the expected orientation of x, and set the fixation to this value. Exit when one of the following occurs:
   (a) If an object is detected along the TZ (tracking zone) that matches x, set the fixation of the camera to the relative orientation of the object and exit via f-in-TZ (found in TZ);
   (b) If an object that matches an anchor other than x is detected along the TZ, exit via nf-in-TZ (not found in TZ);
   (c) If the physical scan is completed, exit via the SC (scan complete).
3. LookTo: turn the camera to the current fixation:
   (a) If an object that matches x appears within the TZ, exit via f-in-TZ;
   (b) If an object that matches an anchor other than x appears within TZ, exit via nf-in-TZ;
   (c) When the desired orientation of the camera has been achieved and no object is detected that matches x, exit via PA (position achieved).
4. Anchor: measure the (relative) orientation and distance of the object and update the x anchor associated with the object; and select a new focus.
5. Serendipitous anchoring: if an object other than the one represented by x is in TZ, measure its (relative) orientation and distance and update the corresponding anchor; and continue to search x.
6. Lost or Occluded: if either Scan or LookTo has been completed without finding the object that x represents, mark x as lost and proceed via LG to select a new focus of attention.

3. COOPERATIVE ROBOT PLATFORMS

The kinematic models of two mobile robots shown in Figs. 5 and 6 are for the follower robot and the leader robot respectively. The motion of the robot's body is controlled by a differential wheel drive. Much of the platform is tailored from the system introduced by [9,10] with minor changes. Here, the main difference between the follower and the leader robots is in the construction of the hands. The follower robot's hand is flexible along its length such that its length can stretch or shrink, while the leader robot's hand is rigid. Both hands are assumed to be firmly hooked with the cargo. Moreover, contrary to the system introduced in [9,10], here, both hands are not actuated and can freely rotate along the reference point O. This means that the orientation and position of the cargo will depend on the position of the two cooperative robots and the distance between them. Thus, the length of the follower's hand depends only on the distance between the reference points of the two cooperative robots.

Both the leader and follower robots are equipped with panned cameras as the primary external sensors. For simplicity, the sensor region is modeled as a triangle, in which R represents the range of the sensor, ρ represents the field of view of the sensor and ω_c represents the speed of panning the camera. Once an object's reference point is within the area of the triangle, the sensor is assumed to be able to sense and retrieve information regarding that object. In the event that two objects are inside the triangle area, the one closest to the observing robot is assumed to be perceivable and the other one is not; this event is called occlusion. Furthermore, β and I represent

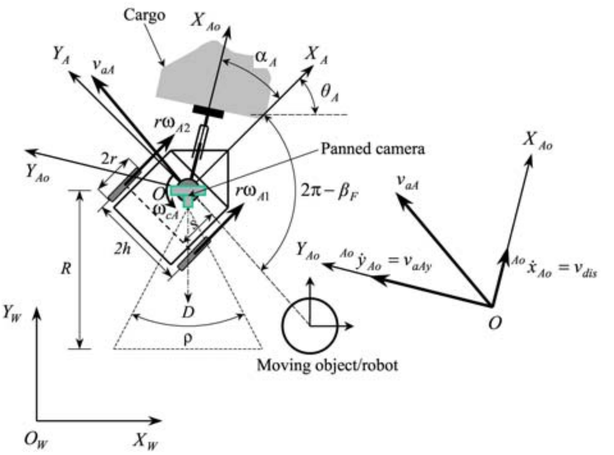


Fig. 5. Kinematic model of the follower robot.

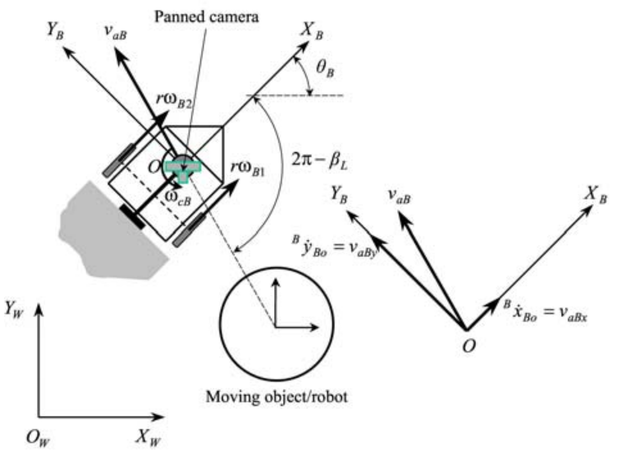


Fig. 6. Kinematic model of the leader robot.

the relative orientation and relative distance from the observing robot to an object, respectively.

3.1. Kinematic model

Local coordinate systems $\Sigma_A(O - X_A Y_A)$ and $\Sigma_B(O - X_B Y_B)$ are set fixed to the frames of the follower and leader robots respectively. Let ${}^A\dot{x}_{Ao} = [{}^A\dot{x}_{Ao}, {}^A\dot{y}_{Ao}]^T$ represent the motion of the follower robot in Σ_A and similarly, let ${}^B\dot{x}_{Bo} = [{}^B\dot{x}_{Bo}, {}^B\dot{y}_{Bo}]^T$ represent the motion along space Σ_B for the leader robot. We define $\omega_A = [\omega_{A1}, \omega_{A2}]^T$ and $\omega_B = [\omega_{B1}, \omega_{B2}]^T$ as the angular velocities of the wheels of the follower and leader robots respectively. The kinematic equations for the follower robot are shown below:

$$\omega_A = A_A^{-1A}\dot{x}_{Ao},$$
$$A_A^{-1} = \begin{bmatrix} 1 & h/s \\ 1 & -h/s \end{bmatrix}, \tag{2}$$

where $2h$ is the tread and s is the offset of the steering axis from the axle of the wheel.

Equations for the leader robot can be derived easily in the form similar to the above equations. Similar to [10], an additional coordinate system $\Sigma_{Ao}(O - X_{Ao} Y_{Ao})$, is set fixed to the hand and point O of the follower robot and the motion along this space is given as ${}^{Ao}\dot{x}_{Ao} = [{}^{Ao}\dot{x}_{Ao}, {}^{Ao}\dot{y}_{Ao}]^T$. This additional frame is used for generating cooperation and avoidance control for the follower robot. Transforming of the motion from Σ_{Ao} space to Σ_A space is performed according to:

$$^{A}\dot{x}_{Ao} = {}^{A}_{Ao}R\,{}^{Ao}\dot{x}_{Ao},$$
$${}^{A}_{Ao}R = \begin{bmatrix} \cos(\alpha_A) & -\sin(\alpha_A) \\ \sin(\alpha_A) & \cos(\alpha_A) \end{bmatrix}, \tag{3}$$

where α_A denotes the angle between X_A and X_{Ao} axes.

3.2. Sub-controllers

Although it is not our primary goal to introduce a novel control system for the cooperative platform considered here, it is still necessary for us to design the inner makings of the NCM as well as the panning control system in the PSM so that we can test the system in simulation. Thus, the author considers only simple controllers and makes no claims of the novelty and efficiency of such controllers. Figs. 7 and 8 show the block diagrams of control systems for the follower and leader robots respectively. In the latter, the controller is composed of two sub-components, namely the obstacle avoidance controller and the trajectory following controller. At any given time, only one of these two sub-controllers will be active; it will switch between sub-controllers depending on the situation. Alternatively, the follower robot's controller is composed of an obstacle avoidance sub-controller and a hand-length controller.

3.2.1 Obstacle avoidance

We use a Mamdani type fuzzy logic (FL) controller [24] for the leader's obstacle avoidance. The controller basically generates the leader avoidance vector, v_{aB}. This vector is parallel and in-line to the segment connecting O and the perceived location of the reference point of an obstacle. The antecedent part of the fuzzy rules takes the membership function for the relative distance I (see Fig. 9) to an obstacle. The leader avoidance vector v_{aB} is transformed into x and y components along the space Σ_B (see Fig. 6). Accordingly, these vector components will dictate

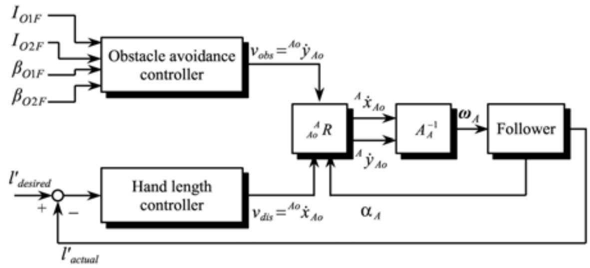


Fig. 7. Follower robot controller module.

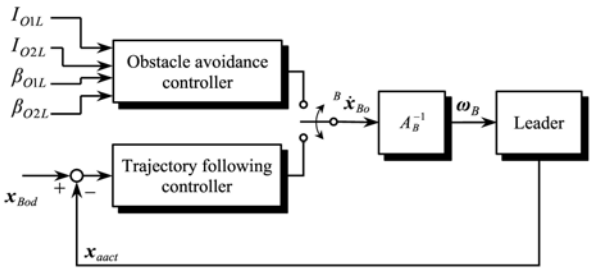


Fig. 8. Leader robot controller module.

the motion of the leader robot. Similarly, the obstacle avoidance controller of the follower robot generates an avoidance vector, v_{aA} as shown in Fig. 5. This vector is also resolved into x and y components along the coordinate Σ_{Ao}. But here, only the x component is used to generate the avoidance motion because the y component will cause an error to occur with the effort of the hand-length controller. Moreover, the FL-based controller output is generated using a centroidal calculation, which returns the center of area (COA) under the curve of a membership function. The COA can be computed as

$$COA = \frac{\sum_{i=1}^{n} A_i \overline{x}_i}{\sum_{i=1}^{n} A_i}, \tag{4}$$

where A_i is the area of the triangle i, $\overline{x}_i$ is its COA, and n is the number of rules.

3.2.2 Hand controller

The hand-length controller of the follower robot produces the hand-length control vector, v_{dis} along the x axis of Σ_{Ao}. Similarly, the hand-length controller consists of a Mamdani type FL controller [24], where the antecedent part of the rules has two parts, taking the memberships for the length error $e_l = l'_{desired} - l'_{actual}$ and the length error rate $\dot{e}_l$ (see Fig. 10), where $l'_{desired}$ and l'_{actual} denote the desired and actual length of the follower robot's hand

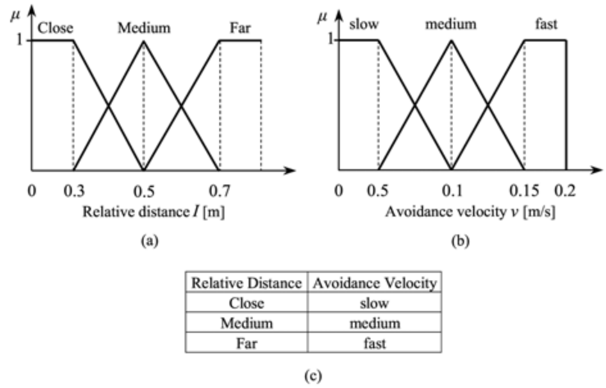


Fig. 9. Membership functions and rules for the avoidance behavior.

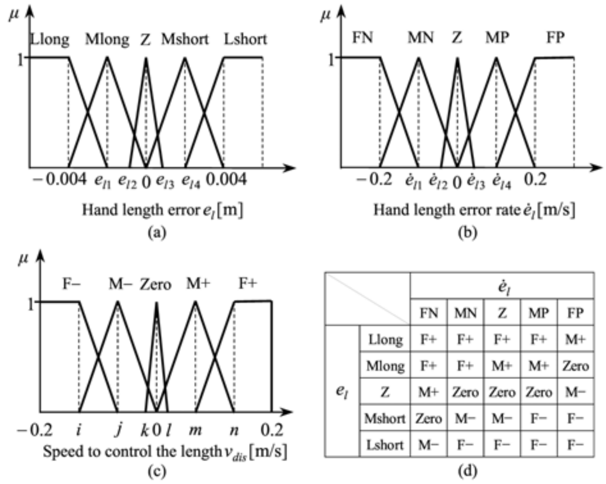


Fig. 10. Membership functions and rules for the hand-length controller.

respectively. The two inputs are combined using the AND operator and the crisp output is generated using centroidal calculation. Fig. 10 shows further details; (a) and (b) indicate the input membership functions for e_l and $\dot{e}_l$ respectively; (c) shows the output membership function for v_{dis}; and (d) displays the rules of the controller. For the hand length error e_l, five memberships are assigned, such that $e_{l1} = -0.002$, $e_{l2} = -0.0015$, $e_{l3} = 0.0015$, and $e_{l4} = 0.002$. A negative value of e_l tells that the current length of the hand is longer than the desired value; "Llong" represents the higher end of being long, "Mlong" signifies medium long and "Z" stands for zero. On the other hand, a positive e_l tells that the current length of the hand is shorter than the desired value, "Mshort" stands for medium short and "Lshort" is the higher end of being short. For the rate $\dot{e}_l$, five memberships are assigned with $\dot{e}_{l1} = -0.1$,

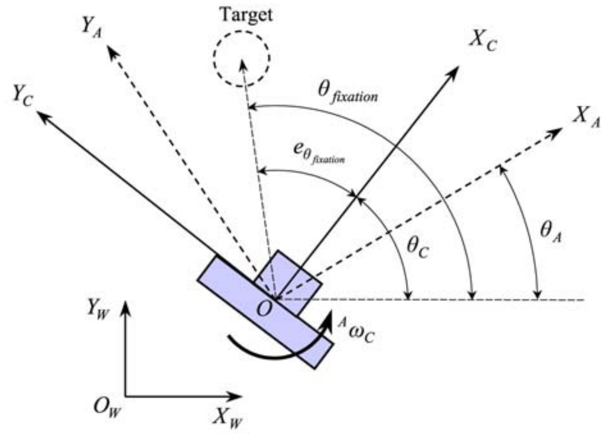
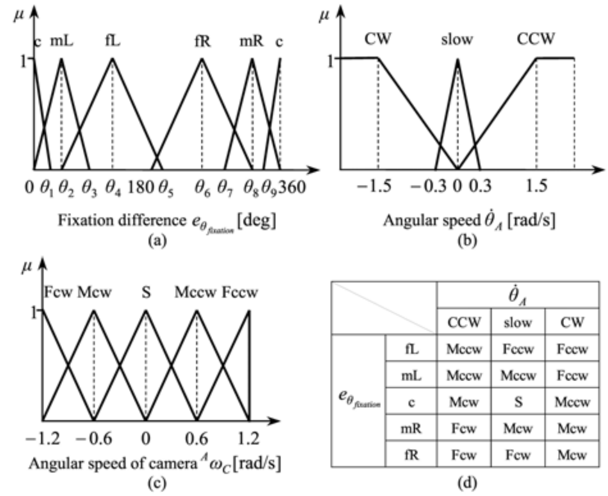


Fig. 11. Kinematic parameters to control the follower robot's panned camera.



		$\dot{\theta}_A$		
		CCW	slow	CW
$e_{\theta_{fixation}}$	fL	Mccw	Fccw	Fccw
	mL	Mccw	Mccw	Fccw
	c	Mcw	S	Mccw
	mR	Fcw	Mcw	Mcw
	fR	Fcw	Fcw	Mcw

Fig. 12. Membership functions and rules for sensor control.

$\dot{e}_{l2} = -0.025$, $\dot{e}_{l3} = 0.025$, and $\dot{e}_{l4} = 0.1$. The memberships are labeled accordingly; "FN" means fast negative, "MN" stands for medium negative, "Z" signifies zero, "MP" indicates s medium positive, and "FP" means fast positive. Conversely, five memberships for the output are assigned with $i = -0.1334$, $j = -0.0667$, $k = -0.0167$, $l = 0.0167$, $m = 0.0667$, and $n = 0.1334$. The output memberships are labeled as follows; "F – " for fast but negative, "M – " indicates medium speed but negative, "Zero" stands for zero speed, "M+" means medium speed but positive, and "F+" for fast and positive.

3.2.3 Sensor motion control

Due to the similarity of functions, the controllers for the two cooperative robot's panned cameras were constructed in the same manner. As with the hand-length controller, the author used a Mamdani type FL [24] to produce a coordinated panning action for each camera of the follower and leader robots.

Shown in Fig. 11 are the kinematic parameters of the follower robot that are relevant for controlling its panned camera. Attached to the frame of the camera is a local coordinate space in which the axis X_C is in-line with the line of sight of the camera. Based on Σ_C, the orientation of the perceived or believed location of a fixation target is denoted as $e_{\theta_{fixation}}$. In particular, this parameter is called the orientation error of the camera. The coordinate Σ_A is also shown to represent the orientation and motion of the main body of the follower robot. The angular speed of the camera relative to the main body of the follower robot is denoted as ${}^A\omega_C$. Moreover, Fig. 12 presents the details of the FL-based panned camera controller, where in (a) the membership function of the error $e_{\theta_{fixation}}$ is shown and (b) depicts the membership function for the main body's angular speed, $\dot{\theta}_A$. Both memberships for $e_{\theta_{fixation}}$ and $\dot{\theta}_A$ are used as inputs in the antecedent part of each rule and combined using the AND fuzzy operator. In (c) the output membership function is shown; the rules will produce values for ${}^A\omega_C$ through centroidal calculation method. Moreover, the input $e_{\theta_{fixation}}$ is divided into 5 memberships such that $\theta_1 = 1.0$, $\theta_2 = 1.5$, $\theta_3 = 3.0$, $\theta_4 = 93.25$, $\theta_5 = 185.0$, $\theta_6 = 269.25$, $\theta_7 = 357.0$, $\theta_8 = 358.5$, and $\theta_9 = 359.0$. The memberships for the input variable $e_{\theta_{fixation}}$ are labeled as follows; "c" means close, "mL" signifies medium left-side, "fL" stands for far left-side, "fR" indicates far right-side, and "mR" means medium right-side. As well, the memberships for the input variable $\dot{\theta}_A$ are labeled as follows; "CW" means clockwise rotation, "slow" indicates slow or no rotation and "CCW" means counter clockwise rotation. Moreover, the output memberships are labeled in this way; "Fcw" means fast clockwise motion, "Mcw" refers to medium clockwise motion, "S" signifies slow, "Mccw" denotes medium counter clockwise motion, and "Fccw" means fast counter clockwise motion. Note here that the target or fixation information comes from either LookTo or the Scan process of the active perceptual anchoring.

3.2.4 Trajectory following

To achieve the goal of transporting the cargo to the target destination, the leader initiates the translational motion to the target. In particular, it is assumed that the leader can readily receive a desired trajectory $x_{Bod} = [x_{Bod}, y_{Bod}]^T$ coming from a hypothetical planner [9, 10]. We simply used a radial basis function neural network (RBFNN) [25] with Gaussian activation function to generate a trajectory following motion. The inputs to the NN are the trajectory following errors, e_x and e_y, their corresponding rate, $\dot{e}_x$ and $\dot{e}_y$, and the leader robot orientation θ_B, where $e_x = x_{Bod} - x_{aact}$ and $e_y = y_{Bod} - y_{aact}$ denote the errors between the desired trajectories (x_{Bod}, y_{Bod}) and the actual leader robot position (x_{aact}, y_{aact}). The RBFNN outputs a velocity vector ${}^B\dot{x}_{Bo}$ along the space Σ_B.

To train the NN, the author used a Genetic Algorithm (GA) [26,27] to minimize the error of following a trajectory. During the GA optimization, each fitness is tested on a training trajectory. The training trajectory has enough complexity so that the outcome of GA will give the RBFNN enough capacity to follow any given trajectory with minimal error. The fitness function is given as:

$$f = \sum_{t=0}^{T_{\max}} W_1 e_x^2 + W_2 e_y^2 + P_{v_{\max}}, \tag{5}$$

where $T_{\max}$ is the amount of time the mobile robot will try to follow the training trajectory, $P_{v_{\max}} = 1$ if ${}^B\dot{x}_{Bo}$ exceeds the linear velocity limit and $P_{v_{\max}} = 0$ otherwise. The weights W_1 and W_2 are introduced to balance the effect of position errors relative to the penalty $P_{v_{\max}}$. Both of the weights, W_1 and W_2 were set empirically at 100. The GA parameters were initialized accordingly: the population size is 100, the crossover rate is 0.6 (uniform), the mutation rate (i.e., probability of mutation) is 1, the selection criteria are based on selecting the best 10 individuals, and the generation time is 15,000.

4. SIMULATION STUDY

We conducted a simulation test to demonstrate how the proposed concept works. We used a small sized version of the cooperative platform discussed in the previous section. The wheel radius is set to 0.065 [m]; the offset distance of the reference point from the wheel axis s, is set to 0.08 [m]; h is 0.06 [m]; the sampling width is set to 0.02 [s]; and the linear velocity of each cooperative robot is limited up to 0.2 [m/s], while the maximum panning velocity for the camera is set to 1.2 [rad/s]. Moreover, aside from the two cooperative robots there are two other robots namely O1 and O2 operating in the same environment. O1 and O2 move along their own trajectories (as shown in Fig. 13) at a speed of 0.15 [m/s].

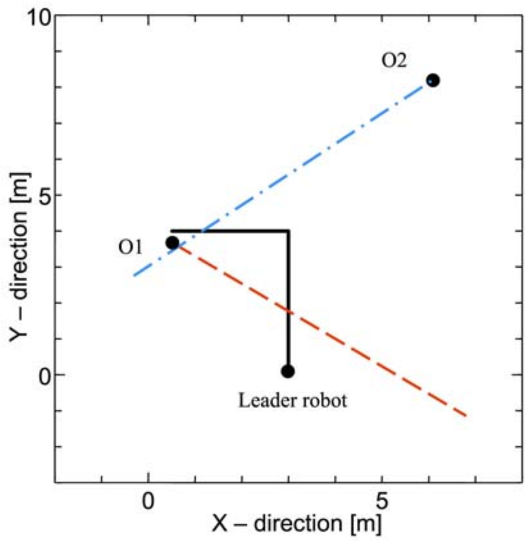


Fig. 13. Desired trajectory for the leader robot and the trajectories for the two other robots working in the environment. Starting positions are marked with a circle.

Other details such as the field of view of the panned camera ρ, is set to 15 [degrees], and the range R is set to 10 [m]; this is sufficient to exclude the range as a problem source. With this, the problem is reduced to a limited field of view and occlusion. The initial length (or desired length) of the follower's hand and the leader's hand is set to $2s$ [m]. We assume a cargo having a square base with size $2s\,[\mathrm{m}] \times 2s\,[\mathrm{m}]$.

The main task of the two cooperative robots is to transport the cargo to the desired location via a predefined trajectory. The trajectories for the leader and the two other robots are shown in Fig. 13. The leader robot task is to follow the trajectory and avoid collisions with other robots working in the same environment. It is assumed that the two other robots are blind so that they don't have any ability to avoid collisions, i.e., they will just go straight and follow their trajectories. In contrast, the follower robot is designed to have the ability for self and cargo preservation, i.e., it is capable of performing collision avoidance not just for its own body but also for the cargo as well. Moreover, the follower robot is designed to cooperatively carry the cargo safely by maintaining a safe hand length. Unlike the case of the leader robot's hand, the hand of the follower robot is capable of increasing and decreasing its length.

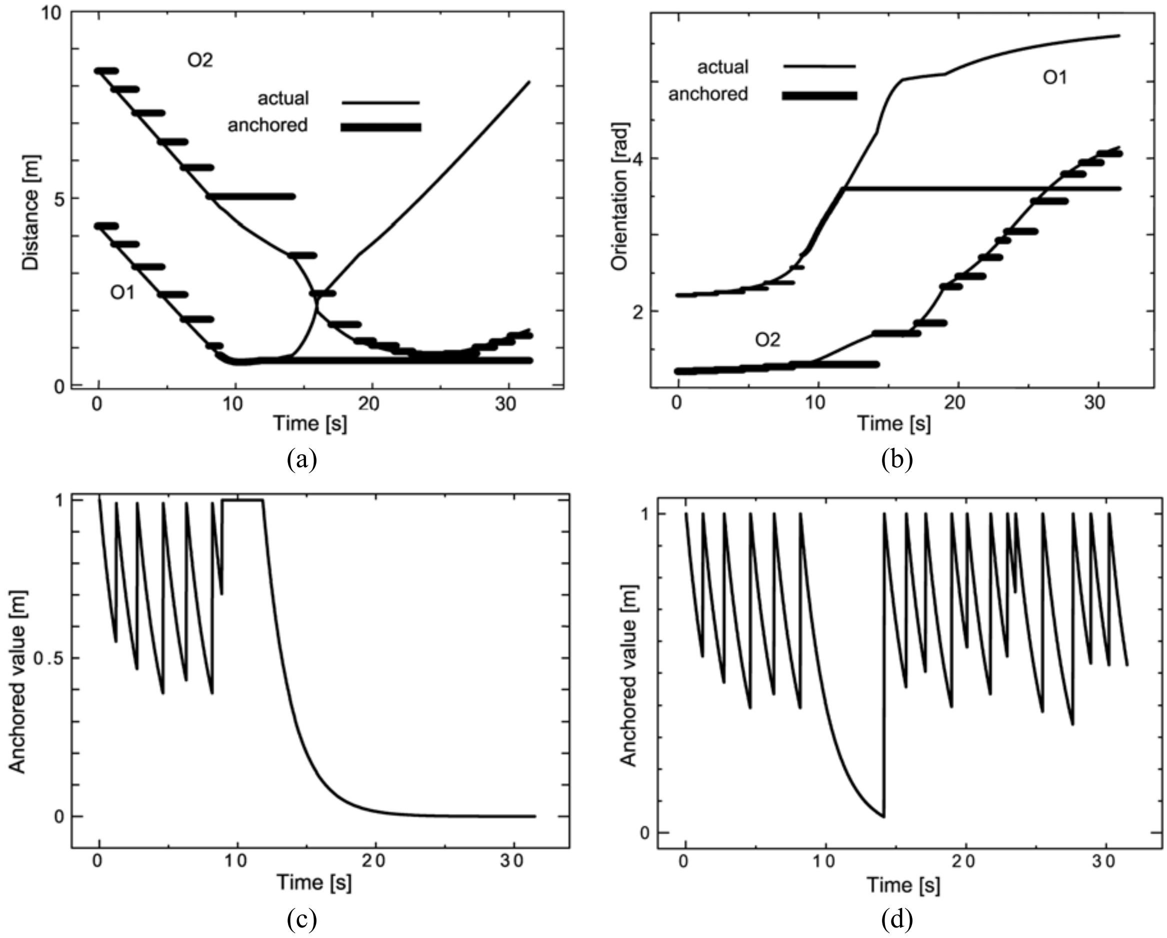


Fig. 14. The leader's eye-view. A and B show the anchored and actual values of distance and orientation. C and D on the other hand show the plot of the anchored measure for O1 and O2, respectively.

5. RESULTS AND DISCUSSION

The simulation results can be shown in two views on how each of the cooperative robots sees and remembers the states of the relevant features in the environment. Information in the anchors and their actual equivalent were plotted against time. At any given time, this information in the anchors is responsible for defining the awareness of each cooperative robot.

With respect to the leader robot's eye-view, the results are shown in Fig. 14, where (a) and (b) indicate the anchored relative distance and orientation for O1 and O2. On the other hand, (c) and (d) show the anchored values for robots O1 and O2 respectively for the entire simulation time. The results demonstrate that before hitting the ninth second mark, the perception system evenly anchored both O1 and O2, that is, the leader robot's attention was alternately shifted to O1 and O2 or the camera swings back and forth between O1 to O2. Soon after O1 came very close, the avoidance module of the leader robot was activated. The activation resulted in the assignment of a higher *needed* value for the anchor O1. This in turn resulted in a full tracking attention for O1; its anchor contents were updated constantly for the time being while O2 was left unattended for approximately 3 seconds. When the leader robot and O1 parted, O1 swung to the south of the leader robot, which later resulted in an occluded view (i.e., the leader robot was unable to visually track O1) due to the presence of the cargo and the follower robot in the direction in which O1 could be found. The anchor for O2 was updated again once the close encounter with O1 was over, and the leader robot could safely track O2.

With respect to the follower robot's eye-view, the results are shown in Fig. 15, where (a) and (b) show the content of the anchors for O1 and O2 respectively for the entire simulation time, and (c) and (d) show the *anchored* values for O1 and O2 respectively. The result tells us a different story with what the leader robot saw. The plots indicate that, at most of the entire simulation times, O2 is not visible for the follower robot. O2 starts to appear only near to the end of the simulation time (i.e., on the 26th seconds mark). This happens because O2 started up from north

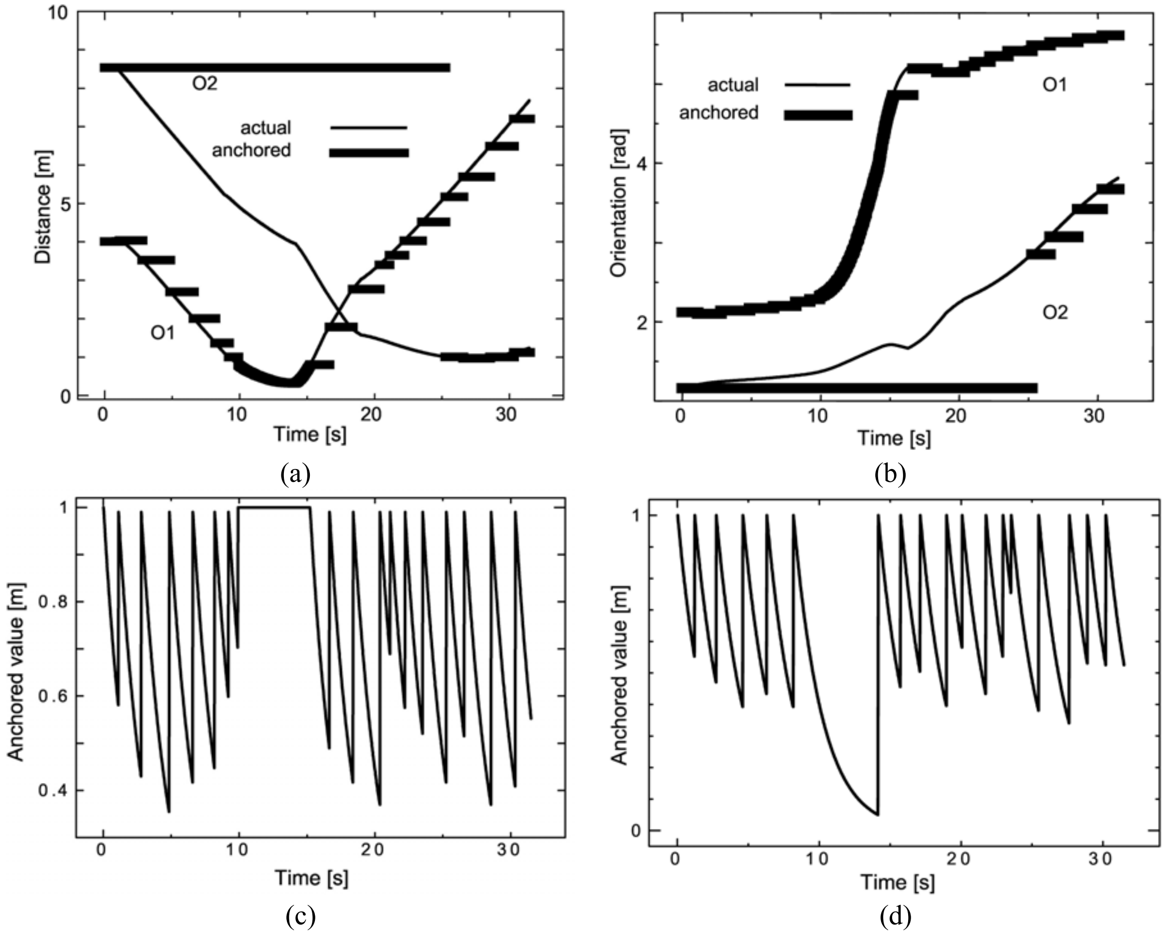


Fig. 15. The follower's eye-view. A and B indicate the anchored and actual values of distance and orientation. C and D on the other hand show the plot of the anchored measure for O1 and O2, respectively.

relative to the place where the follower robot began (see Fig. 13), and because the follower robot follows the leader robot it cannot see up north due to occlusion generated by the presence of the leader robot in that direction. Alternatively, O1 is visible to the follower robot throughout the simulation time due to the fact that no physical object can obstruct the view of the follower robot in any direction along the southern part of the environment.

Here, the two cooperative robots can broadcast to each other their position and the anchors. Specifically, if every time O1 or O2 is lost (or occluded), a cooperative robot will use the information available from the anchor of the other cooperative robot. This technique could result in more highly efficient perception control.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented a formalized approach for controlling the perceptual effort to enhance the awareness of two decentralized mobile robots designed for cooperative transportation of a cargo object, where the two cooperative mobile robots were equipped with panned cameras that had highly limited field of view and thus each of the cooperative agents could only focus on a small fraction of their environment at any given time. In general, this inherent limitation is further aggravated by occlusion; each cooperative agent is unable to see through the cargo and its partner. These problems can severely affect the awareness of each agent and will make the task practically difficult to implement. For each cooperative agent to be aware of the state of its environment, every agent must be able to efficiently control its perceptual effort. Our approach to awareness was based on active perceptual anchoring (APA). Through APA each cooperative agent was able to control its perceptual effort according to the needs of the task at hand. We defined awareness as knowing the position of the other robots in the environment and implemented it through the use of anchors. We demonstrated the approach through a simulation of two cooperative mobile robots that cooperatively transport cargo to a certain destination through a predefined trajectory, in which the two mobile robots cooperatively carrying the cargo move along a trajectory and avoid collisions with other robots while moving towards the target destination. Our simulation results demonstrated that our approach could work and was potentially feasible.

One area in which our approach to awareness for decentralized cooperative mobile robots can be improved is in the implementation of the dynamic part of anchoring. In the present system, the dynamic part of anchoring is not implemented and because of this each cooperative agent has no ability to extrapolate any information from the previously received data. By implementing or adding a dynamic part to anchoring, each agent will be able to simulate and predict the changes in state of one or more objects (or features) while the perception system is busy observing another object in the environment. Additional benefits of this enhancement can be well observed if the number of other robots working in the same environment is much higher than what we have considered here. As the number of robots moving about increases, the number of times that each of these objects will be anchored by a cooperative agent decreases. Therefore it is beneficial to add prediction capabilities in the anchoring process in order to have better awareness in a busier environment.

REFERENCES

[1] L. E. Parker, "Adaptive heterogeneous multi-robot teams," *Neurocomputing*, vol. 28, no. 1-3, pp. 75-92, 1999.

[2] L. E. Parker, "Multi-robot team design for real-world applications," *Distributed Autonomous Robotic Systems 2*, Springer-Verlag, Tokyo, pp. 91-102, 1996.

[3] C. Jennings, D. Murray, and J. J. Little, "Cooperative robot localization with vision-based mapping," *Proc. of ICRA'98*, 1998.

[4] P. U. Lima and L M. Custo'dio, "Artificial intelligence and systems theory applied to cooperative robots: The SocRob project," *Proc. of the Robo'tica 2002 Portuguese Scientific Meeting*, Aveiro, Portugal, 2002.

[5] L. E. Parker, "Toward the automated synthesis of cooperative mobile robot teams," *Proc. of SPIE Mobile Robots XIII*, vol. 3525, pp. 82-93, 2002.

[6] L. E. Parker, "Cooperative robotics for multi-target observation," *Intelligent Automation and Soft Computing*, vol. 5, no. 1, pp. 5-19, 1999.

[7] L. E. Parker, "The effect of action recognition and robot awareness in cooperative robotic teams," *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 212-219, 1995.

[8] C. F. Touzet, "Robot awareness in cooperative mobile robot learning," *Autonomous Robots*, vol. 2, pp. 1-13, 2000.

[9] X. Yang, K. Watanabe, K. Kiguchi, and K. Izumi, "Coordinated transportation of a single object by two nonholonomic mobile robots," *Proc. of The Seventh Int. Symp. on Artificial Life and Robotics*, vol. 2, pp. 417-420, 2002.

[10] X. Yang, K. Watanabe, K. Kiguchi, and K. Izumi, "A decentralized control system for cooperative transportation by multiple nonholonomic mobile robots," *International Journal of Control*, vol. 77, no. 10, pp. 949-963, 2004.

[11] A. Saffiotti and K. LeBlanc, "Active perceptual anchoring of robot behavior in a dynamic environment," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 3796-3802, April 2000.

[12] S. Coradeschi and A. Saffiotti, "Perceptual anchoring: A key concept for plan execution in embedded systems," *Advances in Plan-Based Control of Robotic Agents*, pp. 89-105, 2002.

[13] S. Coradeschi and A. Saffiotti, "Anchoring symbols to sensor data: Preliminary report," *Proc. of the 17th National Conf. on AI (AAAI)*, pp. 129-135, 2000.

[14] S. Coradeschi and A. Saffiotti (Eds), "Perceptual anchoring: Anchoring symbols to sensor data in single and multiple robot system," *Robotics and Autonomous Systems*, vol. 43, no. 2-3, 2003.

[15] R. C. Arkin, "The impact of cybernetics on the design of a mobile robot system: A case study," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, no. 6, pp. 1245-1257, 1990.

[16] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.

[17] H. Hexmoor, J. Lammens, and S. Shapiro, "An autonomous agent architecture for integrating perception and acting with grounded embodied symbolic reasoning," *Technical Report 92-21, University of Buffalo*, 1992.

[18] A. Chella, M. Frixione, and S. Gaglio, "Conceptual spaces for computer vision representation," *Artificial Intelligence Review*, vol. 16, pp. 137-152, 2001.

[19] A. Chella, M. Frixione, and S. Gaglio, "An architecture for autonomous agents exploiting conceptual representations," *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 231-240, 1998.

[20] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 966-1005, 1988.

[21] P.-E. Forsse'n, "Autonomous navigation using active perception," *LiTH-ISY-R-2395*, *Dept. Electrical Engineering,* Link¨oping University, Sweden, 2001.

[22] J. Miura, H. Kawarabayashi, M. Watanabe, T. Tanaka, M. Asada, and Y. Shirai, "Tracking a moving object by an active vision system: Panther-vz," *Proc. of the Int. Symp. of Robotics, Mechatronics and Manufacturing Systems*,

pp. 957-962, 1992.
[23] R. Johansson and N. Xiong, "Perception management: An emerging concept for information fusion," *Information Fusion*, vol. 4, no. 3, pp. 231-234, 2003.
[24] L. Zadeh, "Advanced fuzzy theory," *Journal of Information and Control*, vol. 8, pp. 338-353, 1965.
[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, Upper Saddle River, NJ, 1999.
[26] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205-230, 1999.
[27] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1-16, 1995.

Sherwin A. Guirnaldo received the B.S. degree in Mechanical Engineering in 1995 from Mindanao State University, Philippines. In 2000, he received the M.S. degree in Computer Application from Mindanao State University - Iligan Institute of Technology, Philippines and in September 2004, he received the Ph.D. degree at Saga University, Japan. Presently, he is a faculty member of the Mechanical Engineering Department of Mindanao State University, Marawi City, Philippines. His research interests include perception control, intelligent control, intelligent systems and artificial intelligence in general.

Kiyotaka Izumi received the B.E. degree in Electronic Engineering from the Nagasaki Institute of Applied Science in 1991, the M.E. degree in Electrical Engineering from Saga University in 1993, and the D.E. degree from the Faculty of Engineering Systems and Technology Department at Saga University in 1996. From April in 1996 to March of 2001, he was a Research Associate in the Department of Mechanical Engineering at Saga University.

As of April 2001, he has been with the Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University. His research interests include robust control, fuzzy control, behavior-based control, genetic algorithms, evolutionary strategy and their applications to the robot control.

Keigo Watanabe received the B.S. and M.S. degrees in Mechanical Engineering from the University of Tokushima, Tokushima, in 1976 and 1978, respectively and the D.E. degree in Aeronautical Engineering from Kyushu University in 1984. From 1980 to March of 1985, he was a Research Associate at Kyushu University. From 1985 to March of 1990, he was an Associate Professor in the College of Engineering, Shizuoka University.

From April in 1990 to March of 1993, he was an Associate Professor at Saga University after which he became a Full Professor in the Department of Mechanical Engineering, Saga University. In addition, as of April 1998, he also became a Full Professor in the Department of Advanced Systems Control Engineering at the Graduate School of Science and Engineering at the same university. He has published more than 440 technical papers in transactions, journals, and international conference proceedings, and is the author or editor of 20 books, including Adaptive Estimation and Control (Englewood Cliffs, NJ: Prentice-Hall, 1991), Stochastic Large-Scale Engineering Systems (New York: Marcel Dekker, 1992), Intelligent Control Based on Flexible Neural Networks (Norwell, MA: Kluwer, 1999) and Evolutionary Computations: New Algorithms and Their Applications to Evolutionary Robots (Heidelberg, Germany: Springer-Verlag, 2004). He is an Active Reviewer of many journals and transactions, an Editor-in-Chief of Machine Intelligence and Robotic Control, and an Editorial Board Member of the Journal of Intelligent and Robotic Systems. His research interests include stochastic adaptive estimation and control, robust control, neural network control, fuzzy control, and genetic algorithms and their applications to machine intelligence and robotic control.