Towards intelligent robotic agents for cooperative tasks ⋆

F. Mirelez-Delgado [∗,∗∗] **A. Morales-Díaz** [∗] **R. Ríos-Cabrera** [∗]
H. Gutiérrez-Flores [∗]

[∗] *Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional Unidad Saltillo*
Ramos Arizpe, Coah. 25900, México
Phone number:+52-844-4389600 Ext. 8500
[∗∗] *e-mail: flabio.mirelez@cinvestav.mx*

Abstract: This paper presents an architecture for two intelligent robotic agents built to perform cooperative tasks. In order to test the agents, it is presented a methodology that allows two omnidirectional mobile manipulators to cooperate in a common task. In the experiments the robots successfully collect a workpiece where one robot picks it up, and hand it over to another robot to complete the sequence. The handing over of the workpiece is done on the fly without resting the object on the floor or any other surface. To achieve the task, the proposal is based on visual servoing, a wireless communication scheme between the robots, two RGB-D sensors and one laser sensor. Through experimental validation, promising results are presented, showing that the robotic agents, can be used in future applications in real world scenarios.

Palabras clave: Autonomous industrial mobile manipulators, sensor networks and wireless communication in manufacturing, smart manufacturing systems.

1. INTRODUCTION

Industrial robots are mainly manipulator arms fixed on the ground to perform dumb, dangerous, dull, dirty and overall repetitive tasks. Contrary to the traditional stationary and pre-programmed production robots, Autonomous Industrial Mobile Manipulators (AIMM) can provide assistance at multiple locations (Madsen et al. (2015)). Basically, a mobile manipulator is a stationary manipulator mounted on a mobile robot so that the locomotion and manipulation tasks may be performed simultaneously. These capabilities give the mobile manipulator advantages over stationary ones, like a bigger task space and a greater autonomy.

The task performed by an AIMM includes feeding raw material to workstations, sub-assembly work, goods transportation, and status monitoring of machines. Of all these applications, transportation and loading seem to be the ones with the most potential for the short term actions, according to Hvilshøj et al. (2009).

The autonomy of a mobile manipulator increases if the robot is equipped with more sensors, for example a camera. With a simple camera, the mobile manipulator can have a better understanding of its environment, but it is necessary to establish methodologies and algorithms to ensure a correct interaction between the robot and the objects of interest, or even with other systems in manufacturing plants.

The interest in vision-based robotic systems for monitoring has increased in recent years due to the tendency of reduced costs cameras and in general all the associated processing systems (Hutchinson et al. (1996)). According to Corke (1996) the visual control has matured quickly and has been applied to robot manipulators as it is based on the visual perception of the robot and the location of a piece of interest.

In Weiss et al. (1987) and Hutchinson et al. (1996), a classification of systems structures with visual feedback was introduced. Manipulating objects with robotic systems is a task that has received much attention from the international scientific community, one of the main challenges is to enable systems with knowledge of the object to be handled. This information includes, location, pose, points for grasping and even if the material is deformable or rigid. Object manipulation with robotic systems is introduced in the work of Murray et al. (1994) and an extensive research on servovisual control manipulation is presented in Kragic and Christensen (2002) and Corke (1996).

This paper introduces an architecture for intelligent robotic agents, based on a methodology that aims to carry out cooperation between mobile manipulators to transport workpieces from one location to another, having as a constrain that the object must be handed over on the fly by one robot to the other. In the experiments we use two Kuka youBots equipped with an RGB-D sensor and one robot also has a laser sensor. An efficient algorithm for object segmentation is applied and a wireless communication system between the robots is also developed.

Section 2 shows the related work about Kuka youBot and the AIMM in an industrial scenario. The kinematic model for the youBot platform and its arm and the image

⋆ The first author thanks to CONACyT for the PhD scolarship no. 395788.

processing for the object identification are presented in Section 2. The control law and the algorithm that allows the youBots to cooperate are described in Section 3. The validation for this proposal is presented in Section 4 and finally Section 5 provides the conclusion.

2. PRELIMINARIES

The Kuka youBot is an omnidirectional mobile manipulator specially developed for academic and research purposes Bischoff et al. (2011). The arm direct kinematics can be found clearly expresed in Dwiputra et al. (2014) where Modelica software is used for the arm characterization. Also in Sharma et al. (2012) a unified inverse kinematic for Kuka youBot is established allowing to resolve efficiently the singularities of the whole kinematic model.

A well documented work is presented in Keiser (2013), where the author implemented a torque control at the youBot arm in order to pickup objects from the ground. The objects identification is acomplish by using an RGB-D sensor and the Point Cloud Library. The RBG-D sensor is fixed in a supporting structure located on the back of the mobile platform, allowing the objects to be appeared inside the sensor's field of view. However the arm control was implemented as an open loop, it means that the arm's trajectory is computed offline.

In Schoen and Rus (2013); Dogar et al. (2014); Wang et al. (2013); Hekmatfar et al. (2014) one or more groups of youBots are used in order to manipulate larger objects using wide variety of sensors and algorithms like collision avoidance, interconnected planners or adaptive planning.

Regarding real manufacturing plants only few works have been implemented. Some of them are the "little helper" and the "onmiRob" that work together in a real industrial scenario. The first one performs logistics of individual parts, and the second one works at an assembly station (Bøgh et al. (2012)). In Madsen et al. (2015) and Hvilshøj et al. (2012) the "little helper" is used at a manufacturing plant as a multiple-part feeder. Finally Dang et al. (2013) presents a real-world implementation of an AIMM, taking into to account schedules and real-time.

2.1 Robotic Agent Kinematics

The Kuka youBot's mobile base has four Mecanum wheels that allow free displacement in the Cartesian plane. However, in this kind of system it is necesary to consider the rollers disposition inside the Mecanum wheel to obtain a better kinematic model. According to Nagatani et al. (2000), the kinematic configuration of an omnidirectional mobile robot is given by the set of equations (1).

$$
\begin{aligned}
\dot{x} &= v_l = \frac{1}{4}(d_{b1} + d_{b2} + d_{b3} + d_{b4}) \\
\dot{y} &= v_t = \frac{1}{4}(-d_{b1} + d_{b2} + d_{b3} - d_{b4}) \tan(\alpha_b) \\
\dot{\theta}_b &= v_a = \omega = \frac{1}{4}(d_{b1} + d_{b2} + d_{b3} + d_{b4})\beta
\end{aligned}
\tag{1}
$$

where d_{bi} represents the linear velocity of each wheel of the platform. The variables v_t, v_l and v_a are the transversal, longitudinal and angular velocities respectively. The α_b and β parameters are computed experimentally because they depend of the roller's angle inside the Mecanum wheel.

The mobile robot movement is given by the combination of the wheel's linear velocities. For example, the transversal traslation and rotation movements are depicted in Figure 1, where the faded platform shows where the robot will be at a certain time.

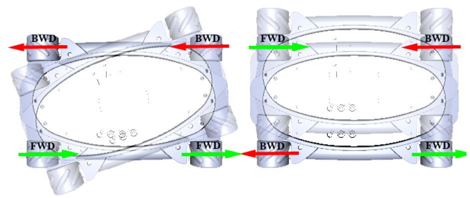

Fig. 1. Mobile platform movement.

The Kuka youBot arm is a manipulator with five degrees of freedom (all of them are rotational). The distance between the body and their respective joint limits can be seen in Figure 2. The end effector is a two fingers gripper that allows the grasping and manipulation of small objects.

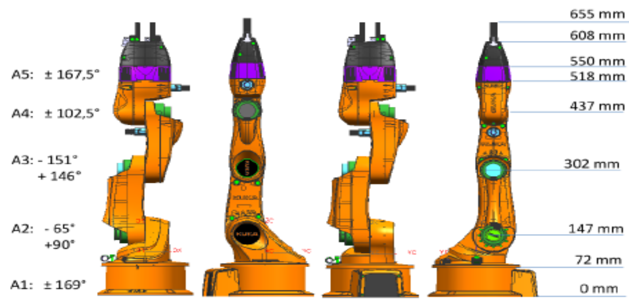

Fig. 2. Kuka youBot arm manipulator specifications.

The kinematic model for the youBot arm is obtained by using the Denavit-Hartenber algorithm, see Lipkin (2005). The referential frames for each joint can be appreciated in the Figure 3. The Denavit-Hartenberg parameters for the manipulator are shown in the Table 1, whereas Table 2 contains the distances and angles of the referential frames. The parameters of Tables 1 and 2 allow the computation of the homogeneous transformation matrix described in equation (2).

Table 1. Denavit-Hartenberg parameters for youBot manipulator.

Body	$\boldsymbol{\theta}$	**d**	**a**	$\boldsymbol{\alpha}$
1	q_1	0.147	0.0330	$\frac{\pi}{2}$
2	q_2	0	0.1550	0
3	q_3	0	0.1350	0
4	$q_4 + \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
5	q_5	0.2175	0	0

$$
A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & -S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

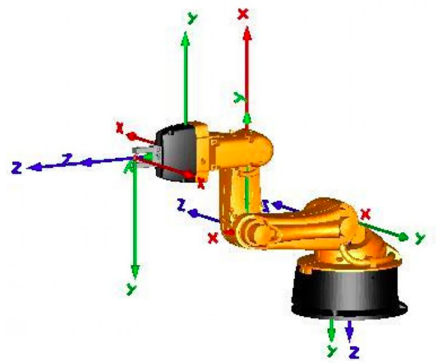

Fig. 3. Referential frame position for each joint.

Table 2. Kinematic chain for the youBot manipulator.

	Previous frame	Traslation [cm]			Rotation [degrees]		
		x	y	z	x	y	z
Joint 1	Base	2.4	0	11.5	180	0	0
Joint 2	Joint 1	3.3	0	0	90	0	-90
Joint 3	Joint 2	15.5	0	0	0	0	-90
Joint 4	Joint 3	0	13.5	0	0	0	0
Joint 5	Joint 4	0	11.36	0	-90	0	0
Gripper	Joint 5	0	0	5.716	90	0	180

Where S and C are the sine and cosine functions respectively. The total transformation matrix is given by the successive multiplication of each homogeneous transformation.

$$T_n^0 = A_1^0 \dots A_n^{n-1} \tag{3}$$

2.2 Image processing

The Asus Xtion Pro (see Figure 4) is capable of obtaining a color image (RGB) and a matrix (image) with depth values. With those two images a point cloud can be obtained. The main advantage of using this kind of sensors is that they can be programmed using open source codes such as OpenNI and OpenCV.

Fig. 4. RGB-D sensor characteristics.

The procedure used in this paper to extract an object of interest from an RGB image consist of converting this RGB image to a HSV image. This is because the HSV color space is more robust to illumination changes than the RGB color space. Then the image is processed in order to find all the blue pixels in the scene. The next step is to gather this pixels in a countour, so that the centroid of the group of pixels can be computed (see Figure 5). In the RGB image there is a reference position that the robot needs to align with the object's centroid. In this way, there are two errors (in X and Y image's axis). The X-error and Y-error are the inputs for the Kuka youBot control law.

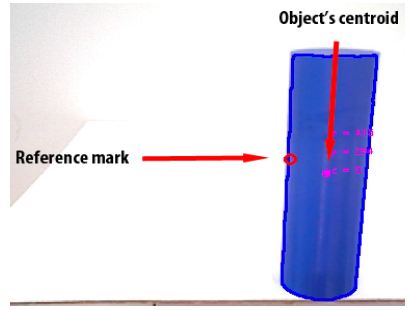

Fig. 5. Image procesing with the Asus Xtion Pro.

The RGB-D camera is used to identify QR codes thanks to the "zbar" library. The camera can obtains the area, centroid and orientation for a QR code. In a similar way to the object segmentation, the RGB-D camera provides the X-error and Y-error when a QR code is present into the camera's field of view.

3. CONTROL LAW

The control law proposed in this paper is an image-based visual servoing with proportional and derivative gains. The errors on the X and Y image's axis are defined as follows:

$$\begin{aligned} e_d^c &= p_2 - p_1 \\ e_x^c &= C_{x_d} - C_x \\ e_y^c &= C_{y_d} - C_y \\ e_z^c &= area_d - area \end{aligned} \tag{4}$$

where p_2 and p_1 are the depth values of two points of interest in the scene. The position $(C_{x_d},\ C_{y_d})$ is a fixed point in the image and it is the reference position where the object's centroid must be align $(C_x,\ C_y)$. In a similar fashion $area_d$ is the desired object's area (in pixels), this value is related to the distance between the object of interest and the robot.

The mobile platform has no motion constraints in the Cartesian plane. The X-error can be minimized with platform's lateral movements. Depending on which robot is moving towards the object, the longitudinal movement of the platform can minimize the Y-error or the area error. The control law can be defined as follows:

$$\begin{aligned} v_t &= K_p(e_x^c) + K_d(\dot{e}_x^c) \\ v_l &= K_p(e_z^c) + K_d(\dot{e}_z^c) \text{ or } v_l = K_p(e_y^c) + K_d(\dot{e}_y^c) \\ v_a &= K_p(e_p^c) + K_d(\dot{e}_p^c) \end{aligned} \tag{5}$$

The variables $K_p,\ K_d > 0$ are the proportional and derivative gains.

3.1 Agent Cooperative algorithms

The method proposed is based on a communication scheme and object identification strategies using a RGB-D sensor to segment the objects by color. It is also implemented a visual servoing control on the youBots in order to successfully accomplish the task.

The cooperative scheme is divided into three stages: In the first one, youBot_1 searches the object of interest using its RGB-D sensor and the Hokuyo laser sensor. Once the object has been found and successfully grasped, youBot_1 searchs youBot_2. When youBot_1 finds youBot_2, it sends a message to the second one with the word "view".

The second part of the algorithm consists of searching the object for the youBot_2. Once the object has been found and grasped by youBot_2, the handing over of the object is carry out: youBot_2 sends a message with the word "grasped" and this is received by youBot_1. YouBot_1 opens its gripper and sends back the message with "left". At the final stage youBot_2 has the object grasped and it starts to look for the desired place to rest the object.

4. EXPERIMENTAL RESULTS

The experimental setting used to validate the algorithm proposed in the previous section is depicted in Figure 6, where the two youBots have a random position. The youBot_1 is the one with the Asus in hand. The youBot_2 carries the Asus sensor in a pedestal. The object of interest is a blue cylinder. Finally the desired place to leave the workpiece and the youBot_2 have QR codes identifiers [1].

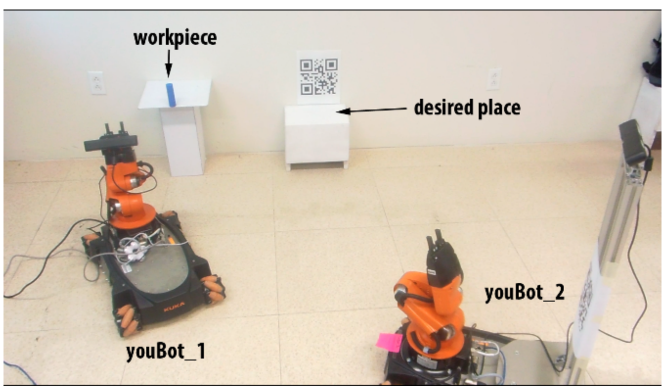

Fig. 6. Experimental setting.

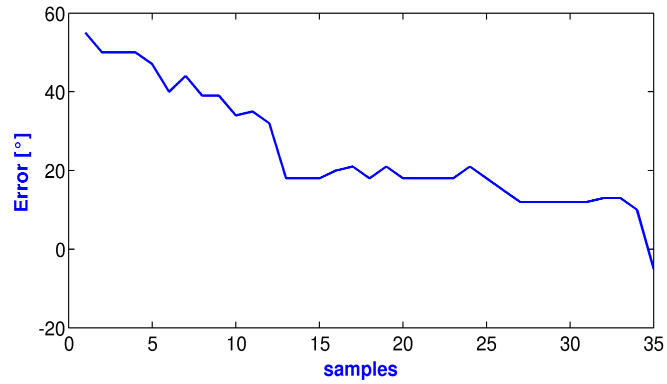

Fig. 7. Orientation error for youBot_1.

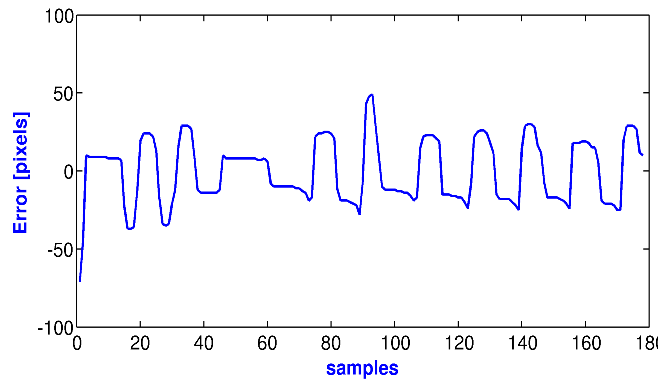

Fig. 8. Lateral error for youBot_1.

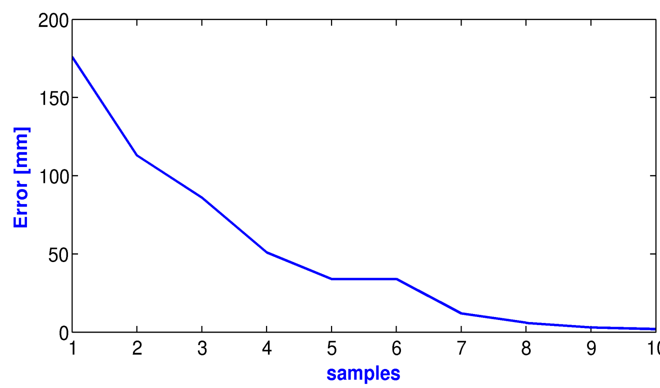

Fig. 9. Longitudinal error for youBot_1.

In the first part of the proccess, the robot aligns perpendicular to the wall. Here it is where the depth values of two different points from the point cloud are taken. Figure 7 shows the youBot_1 orientation error. The next part of the experiment consists of approaching to the object. Figure 8 shows the how image's X-error evolves and Figure 9 shows the distance error between the object and the robot. The behaviour seems to be oscillating, this is because the lateral control action takes no effect when the longitudinal control is applied.

Once the youBot_1 is at 0.1 $[m]$ to the object and the X-error is less than 5 pixels, the manipulator grasps the object with open loop movements. This means that the desired object is in a known position with respect to the robot and the manipulator is moved using inverse kinematic only. After grasping the object, the platform moves backwards until a secure distance is achieved. In the next step, youBot_1 searches youBot_2 by doing rotational movements around his own Z-axis. When the youBot_2's QR code is found in the field of view of youBot_1, an angular control (v_a in (5)) is applied in order to minimized the depth value of two corners of the QR code. This can be appreciated in the Figure 10.

Figure 11 shows the youBot_1's movement during this stage of the experiment. Is important to note that the movement is referenced to its initial positional frame.

Once the youBot_1 is oriented with youBot_2, it sends a message to youBot_2 in order for youBot_2 to begin the search of the object. So the youBot_2 has to minimize the X-error and the Y-error on the RGB image captured by the Asus on the pedestal. The evolution of the two errors mentioned before is shown in Figures 12 and 13.

[1] `https://youtu.be/peAtxUA_tmI`

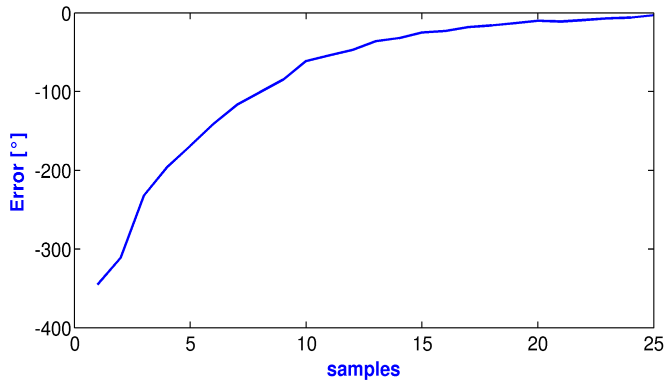

Fig. 10. Orientation error for youBot_1 when searching for youBot_2.

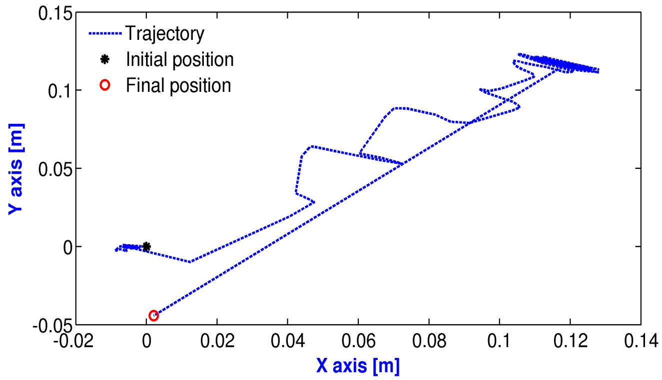

Fig. 11. Trajectory of youBot_1 on the Cartesian plane.

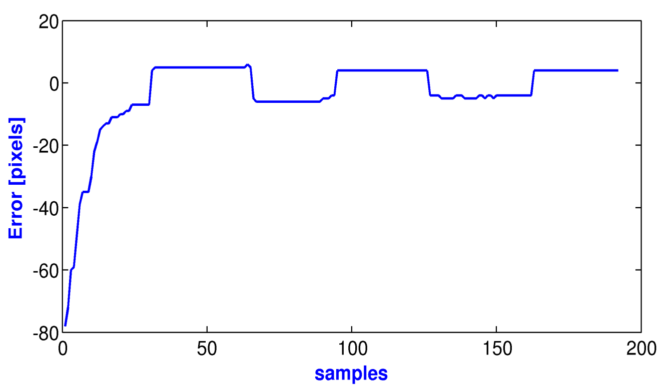

Fig. 12. Lateral error for youBot_2.

When the youBot_2 is close enough to the object being hold by youBot_1, the handing over process begins. The next part of the process involves only the youBot_2. In this stage the robot searches the desired location where the object must be placed. The desired location is identified by QR code. The search starts with youBot_2 spinning around its own Z-axis. When the QR code is found, the angular control (v_a in (5)) is applied again to align the robot with the desired placing location. The orientation error behaviour can be seen in Figure 14.

Once that youBot_2 is aligned, the robot must to approach to the desired place for placing the object of interest. The way to get near to the goal is by using a

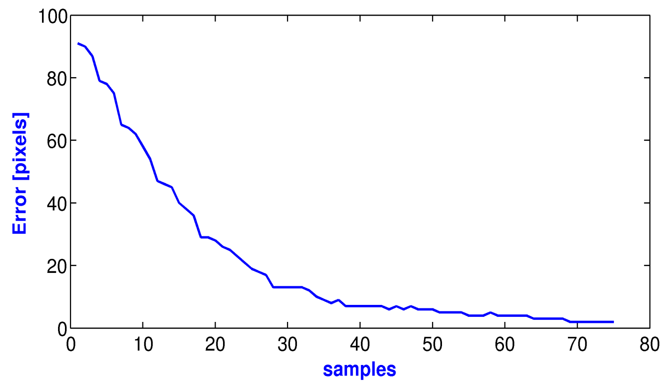

Fig. 13. Longitudinal error for youBot_2.

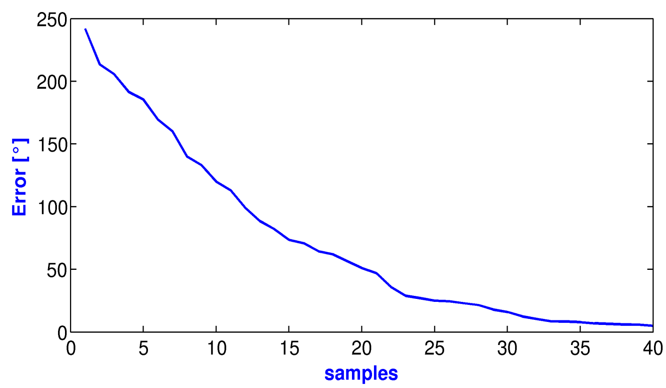

Fig. 14. Orientation error for youBot_2.

lateral and longitudinal control. In this particular case, the youBot_2 moves to the QR code without increases its X-error in the image. For this reason, no lateral control has been applied. The performance of the longitudinal control is shown in the Figure 15.

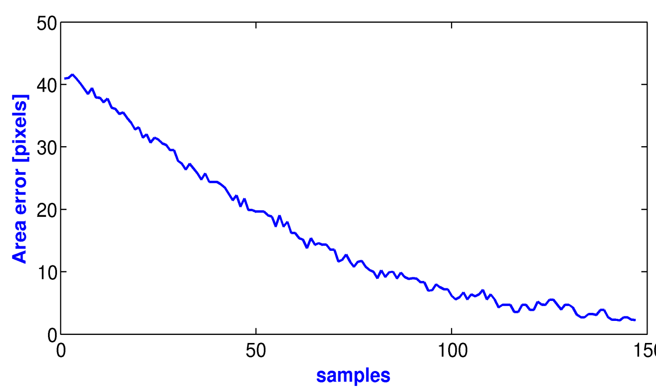

Fig. 15. Longitudinal error for youBot_2.

Finally, in Figure 16 the trajectory for youBot_2 can be seen. In a identical way that youBot_1's trajectory it is represented in the youBot_2's frame.

5. CONCLUSIONS

In this paper a scheme for intelligent cooperative robotic agents was presented. Two mobile robots were equipped with RGB-D sensors to demonstrate the algorithm proposed. A wireless communication scheme was developed

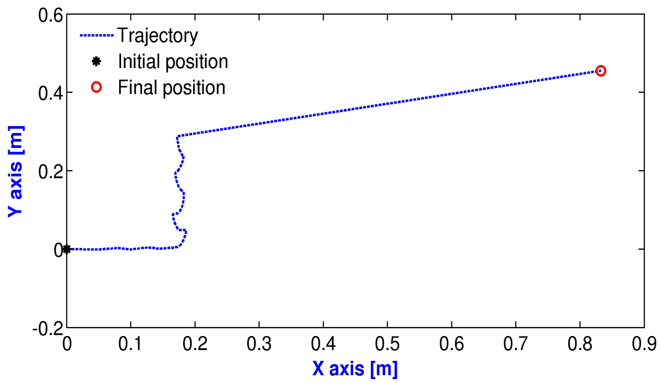

Fig. 16. Trajectory of youBot_2 on the Cartesian plane.

to facilitate the information exchange between the robots by using simple words that allows the beginning of the tasks. In this way, the stages had no time constraints. The main objective of this paper was to identify an object of interest by one robot and to put it in a desired place by a second robot, this goal involved the object handing and the identification of the partner and the desired place. The experimental results show that two robotic agents can cooperate in tasks such as identification and manipulation of workpieces in real world scenarios like warehouses or part supply in assembling tasks. The robots can accomplish their objective without having excessive initial information about the environment. As future work the control law and the image processing algorithms will be improved, including a whole body kinematic model.

REFERENCES

Bischoff, R., Huggenberger, U., and Prassler, E. (2011). Kuka youbot - a mobile manipulator for research and education. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 1–4.

Bøgh, S., Hvilshøj, M., Kristiansen, M., and Madsen, O. (2012). Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (aimm). *The International Journal of Advanced Manufacturing Technology*, 61(5-8), 713–726.

Corke, P. (1996). *Visual control of robots: high-performance visual servoing*. Robotics and mechatronics series. Research Studies Press.

Dang, Q., Nielsen, I., Bøgh, S., and Bocewicz, G. (2013). Modelling and scheduling autonomous mobile robot for a real-world industrial application. In *7th IFAC Conference on Manufacturing Modelling, Management, and Control, MIM 2013, Saint Petersburg, Russia, June 19-21, 2013.*, 2098–2103.

Dogar, M., Knepper, R.A., Spielberg, A., Choi, C., Christensen, H.I., and Rus, D. (2014). Towards coordinated precision assembly with robot teams. *Proceedings of the International Symposium of Experimental Robotics*.

Dwiputra, R., Zakharov, A., Chakirov, R., and Prassler, E. (2014). Modelica model for the youbot manipulator. In *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*, 1205 – 1212.

Hekmatfar, T., Masehian, E., and Mousavi, S. (2014). Cooperative object transportation by multiple mobile manipulators through a hierarchical planning architecture. In *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*, 503–508.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5), 651–670.

Hvilshøj, M., Bgh, S., Madsen, O., and Kristiansen, M. (2009). The mobile robot "little helper": Concepts, ideas and working principles. In *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, 1–4.

Hvilshøj, M., Bøgh, S., Nielsen, O.S., and Madsen, O. (2012). Multiple part feeding – real-world application for mobile manipulators. *Assembly Automation*, 32(1), 62–71.

Keiser, B. (2013). *Torque Control of a KUKA youBot Arm*. Master's thesis, Robotics and Perception Group, University of Zurich.

Kragic, D. and Christensen, H.I. (2002). Survey on visual servoing for manipulation. Technical report, Computational vision and active perception laboratory.

Lipkin, H. (2005). A note on denavit-hartenberg notation in robotics. In *29th Mechanisms and Robotics Conference*, volume 7, 921–926.

Madsen, O., Bøgh, S., Schou, C., Andersen, R.S., Damgaard, J.S., Pedersen, M.R., and Krüger, V. (2015). Integration of mobile manipulators in an industrial production. *Industrial Robot: An International Journal*, 42(1), 11–18.

Murray, R.M., Sastry, S.S., and Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.

Nagatani, K., Tachibana, S., Sofne, M., and Tanaka, Y. (2000). Improvement of odometry for omnidirectional vehicle using optical flow information. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, 468–473 vol.1.

Schoen, T. and Rus, D. (2013). Decentralized robotic assembly with physical ordering and timing constraints. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 5764–5771.

Sharma, S., Kraetzschmar, G.K., Scheurer, C., and Bischoff, R. (2012). Unified closed form inverse kinematics for the kuka youbot. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, 1–6.

Wang, J., Adib, F., Knepper, R., Katabi, D., and Rus, D. (2013). Rf-compass: Robot object manipulation using rfids. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom '13, 3–14. ACM, New York, NY, USA.

Weiss, L., Sanderson, A., and Neuman, C.P. (1987). Dynamic sensor-based control of robots with visual feedback. *Robotics and Automation, IEEE Journal of*, 3(5), 404–417.