Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Assembly/disassembly strategies for service applications

Puente, ST*.Torres, F.* Díaz C.*

**Automatics, Robotics and Computer Vision Group,*
Physics, Systems Engineering and Signal Theory Department
University of Alicante, Spain,(e-mail: Santiago.puente@ ua.es).

Abstract: This paper presents a development of assembly/disassembly systems to be applied in service applications. It uses algorithms developed for industrial applications to perform the service tasks. A robotic system to perform service applications in a cooperative environment with a human is used. The system uses product model to perform the task planning, with the designed task the required grasping points to manipulate the objects are computed according the restrictions. Two experimental results of applying the system to service applications are described.

1. INTRODUCTION

Automation of industrial processes is everyday more defined; nowadays one of the work focuses is the automation of disassembly systems. These systems are highly important to perform an optimal product recycling operations recovering the components and materials that were used in the manufacturing of the products, erasing the necessity of manufacturing these components another time. In this field there are works to reduce the recycling cost and increase the parallelization of the process to obtain the best cost/benefit relation in the recycling process (Lambert, 2007). Nowadays, another of the working fields is the automation of the disassembly task to separate the components of a product using automatic or semi-automatic systems (Kopacek, 2007).

The industrial approach of recycling systems is the initial use of disassembly systems. If a disassembly system is combined with an assembly system a maintenance system can be obtained (Torres and Puente, 2006; Kang and Xirouchakis, 2006) because if there are to substitute a broken component, firsts of all there are to perform a partial disassembly of the product to achieve the broken component, after removing that component, there are to assembly a new one, and finally, there are to re-assembly the components that were removed to achieve the broken component (Giudice and Fargione, 2007)

If the maintenance approach for assembly/disassembly task is taken into consideration, an extrapolation to a new group of applications can be done to the service robots (Garcia *et al.*, 2007). In this field there are great quantities of tasks that are composed by a group of assembly/disassembly actions combined with intelligent manipulation actions.

If the interest is focused in service applications, task like open a closed door by humanoid robot can be considered. In that task, there are to insert the key in the lock, assembly the key. The next step is to turn the key, to open the door, this is an intelligent manipulation. Finally there are to remove the key from the lock, it is a disassembly task.

Once the tasks are defined, there are to plan the actions to be performed for accomplish the target application. The next step has to divide each task in more concrete actions, which can be performed by a robot arm (Velásquez and Nof, 2007). That is possible utilizing a task model, including in the model the components that take part in it and the relation among them (Yu and Li, 2006). There are different types of models, but one useful is one which defines the product like a group of assemblies and sub-assemblies (Torres *et al.*, 2003).

Once the task are planed there are to go further to obtain the grasping points for the components (Díaz *et al.*, 2007). These grasping points must include restrictions based on the specific task that is been performed, not only geometric restrictions based on the components characteristics to be manipulated, because the actions sequence can include manipulation restrictions over some areas of the component. In the key example, if there are to insert the key in the lock the robot can not hold it by the part of the key that it is necessary to insert in the lock.

Sensors are required to perform the described task in a proper way. There are several types of sensor that can be used. For example, visual sensors provide information about the environment and the task that is been performed (Gil *et al.,* 2006). Visual-servoing systems can be used to control a real-time control over the task, especially in the approximation task (Pomares *et al*., 2007). On the other hand, tactile sensors and force sensor provides a precise control in the assembly/disassembly task, because they provide direct information about the environment that is been manipulated and if it is been performed in the correct way (Pomares and Torres, 2005).

Assembly/disassembly task for service applications can be performed putting together all these ideas. In this paper the initial characteristics that have to be taken into consideration to transform from the industrial approach to the service applications are described. Also, several experimental results of the proposed system are described, using an industrial

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.2143

robot arms which allows knowing the results that can be obtained with that approach.

Section 2 describes the developed system. Section 3 describes the system architecture. In Section 4 the experimental results are shown. Finally the Conclusions and the future work are presented in Section 5.

2. DEVELOPED SYSTEM

The service applications can be classified, taking into consideration the type of tasks to be performed, according to four groups, considering that all these type include an assembly or a disassembly tasks. The classification is:

- *Assembly applications*: This type of applications required to assembly a component to achieve the target objective of the application, and can include some type of action with the component once it is assembly with the rest of the system.

- *Disassembly applications*: In this type of applications there are to separate a component from the rest of the product to perform the objective of the application. Once the component is disassembly, the application can required or not to do some action with that component.

- *Assembly/disassembly applications*: This type of tasks required first of all to assembly a component with the whole product, secondly there are to perform and action with the whole product, including the assembly component, and, finally there are to disassembly the component that was previously assembly.

- *Disassembly/assembly applications*: This type of applications required, first of all to remove a component from a whole product, the partial disassembly of the product. Secondly there are to perform and action with the disassembly component or with the rest of the product. Finally there are to assembly the disassembled part of the product to leave it like it was initially.

Taking this classification of service applications into consideration to perform them there are to follow the steps described now:

1. There are to define the application that is going to be performed.

2. There are to create the model of the system. The model provides the relations among components and the assembly order among them.

3. Obtain the group of rules to be followed for a correct performance of the desired application (Torres *et al*., 2003). The whole group of rules can be divided into subgroups according to the type of the application. These sets correspond with the rules for the assembly steps, the rules for the middle of the tasks actions, and, the rules for the disassembly steps.

4. Sort the task to perform in a sequence, taking into consideration the parallelization of some task if there are several robot arms or some humans to interact with the robot arm.

5. Distribute the task among the robot arm and the humans, making sure that the robot or human who has to perform the task has the required tool (Díaz *et al.*, 2006).

6. Using 3D geometric model of the component and the product obtain the grasping points to manipulate the component (Fernandez *et al.,* 2006; Díaz *et al*., 2007).

7. Compute the trajectory to be followed by the robot arm for grasping the component in the desired point and the trajectory required to perform the subsequent action.

8. Perform the action defined in the rule obtained in step 4.

9. If there are to perform more actions before finishing the task return to step 6.

All these steps are controlled by a supervisor system; it is based in artificial vision and force information. The supervisor system allows to control the task that during their execution, making sure that they finishes in the correct way. If some error or problem occur during the execution of one task, the supervisor system must re-planning that task and if it is necessary re-planning the whole application to make sure the system finish as it was expected (Fig. 1).

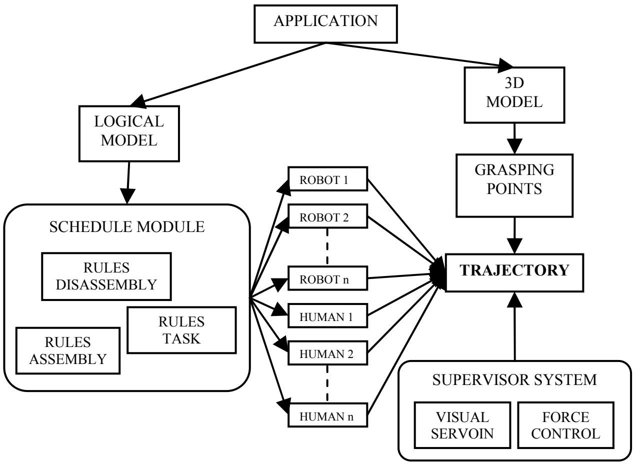

Fig. 1. Assembly/disassembly system flow-chart.

2.1 Skeletisation

The graping point is computed using the skeletisation of one of the inertial planes of the 3D geometric model of the product (Díez *et al.,* 2007). In this subsection how the skeletisation is performed is described.

To calculate a region skeleton thinning algorithms are used. The skeletisation is defined like the process that eliminates boundaries and borders of a planar figure preserving the connectivity without eliminating the ending point until get the figure skeleton, while preserving the structure and the homotopy of the figure that represent a specific object. This approach thins the object to set idealised thin lines which condense the information of the original object (Fig. 2).

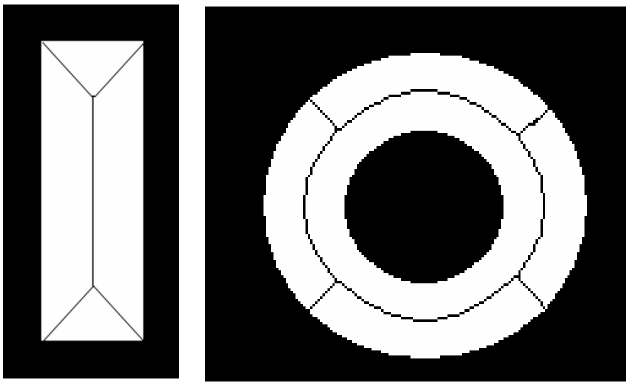

Fig. 2. Skeletons for different figures.

For the Euclidean skeleton, we consider the lattice $P(\Re^2)$, and the families of structuring functions that we use are families of closed balls $B_\lambda(x)$:

$$\forall x \in \Re^2 , \forall \lambda \geq 0 , B_\lambda(x) = \{ y \in \Re^2 , d(x,y) \leq \lambda \} \qquad (1)$$

where d designates the Euclidean distance in the plane. The skeleton seems to be a very natural notion, that does not hurt common sense (Serra, 2002).

These thin lines are also called medial axis transformation (MAT). The skeletisation is based on the concept of maximal disk. Given an interior point to a Euclidean binary image, there exists a larger disk having the point at it centre and also lying within the image (Fig. 3).

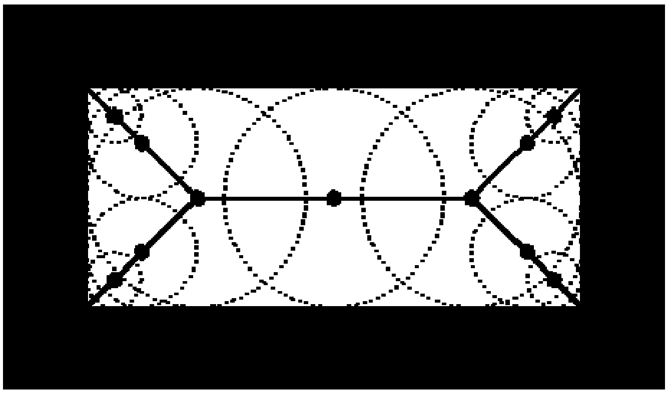

Fig. 3. Skeleton of a rectangle defined in terms of bi-tangent circles.

Regarding the largest disk at a point, there are two possibilities: either there exist another disk lying within the image and properly containing the given disk, or there are does not exist another disk within the image properly contained the given disk. Any disk satisfaction the second condition y called a maximal disk. The canters of all maximal disks comprise the skeleton or medial axis of the image (Dougherty and Lotufo, 2003; Soille P., 1999).

The medial-axis of an area A is a set of pairs defined in equation 2 where B is boundary of the region such that the union of the circles with center x and radius $d_s(x,B)$ is equal to that of region A as shown in Figure 3.

$$\{x, d_s(x,B)\} \, with \; d_s(x,B) = \min\{d(x,z), z \; in \; B\} \qquad (2)$$

The skeleton is useful because it provides a simple and compact representation of a shape that preserves many of the topological and size characteristics of the original shape. Thus, the skeletisation gives a rough idea of the length of a shape by considering just the end points of the skeleton and finding the maximally separated pair of end points on the skeleton (Haralick and Shapiro, 1992).

The skeletons are often very sensitive to small changes in the object. As with thinning, slight irregularities in a boundary will lead to spurious spurs in the final image which may interfere with recognition processes based on the topological properties of the skeleton, that way a 3D CAD model that gives a binary image of the product is use to avoids any kind of errors at the boundaries of the object or any kind of irregularities (Fig. 4).

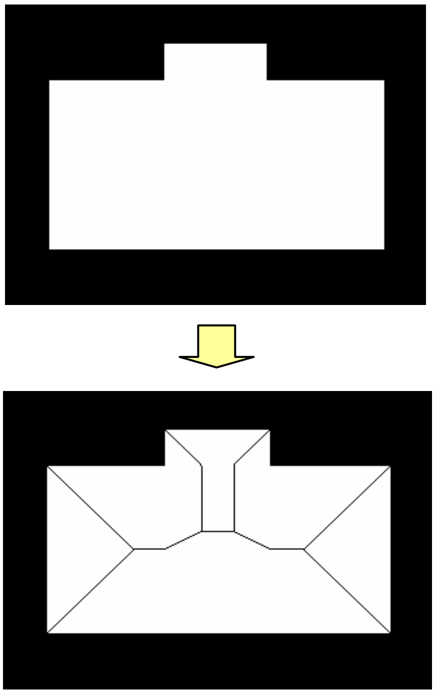

Fig. 4. Skeleton of a rectangle defined in terms of bi-tangent circles.

3. SYSTEM ARCHITECTURE

A robotic working cell has been used to perform the experimental test. The working cell is composed of two industrial robot arm and one human. The first robot is a Mitsubishi ® PA-10 robot arm with seven degrees of freedom with a tool changer at its end and six degrees force sensor. The second robot is Scorbot ER-IX of Intellitek ®, it is a robot arm with five degrees of freedom. The working cell is shown in figure 5.

Inside the robotic working cell, a human can interact with any of the robots and with the product in the middle of the system. Taking that into consideration, the working cell is composed by two robot-arms and the human. As it was said previously the robots can change their tools according the requirements of the task to be performed by them.

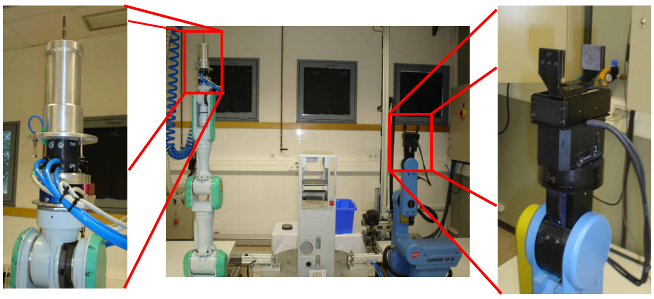

Fig. 5. Robotic working cell.

Added to the robotic part of the working cell, the supervisor system is composed of external stereo vision system, as well as, a camera mount at the end of the robot arm to perform a visual servoing control with eye-in-hand configuration. These vision systems allow controlling how the task is performed.

The system architecture is completed with a force sensor and vision camera for the supervisor sub-sytems. The main characteristic of the JR3 force sensor is that it is capable of measuring the force on three orthogonal axes, and the moment (torque) about each axis. The supervisor sub-systems also use visual servoing. The camera is Photonfocus MV-D752-160-CL-8; it allows capturing 340 fps with 752x582 pixels of resolution.

4. EXPERIMENTAL RESULTS

For testing purposes of the designed algorithms two examples has been make. The first is the key disassembly from a lock, in this example the application, according to the classification performed in the Section 2 is a disassembly application. The second example consist in a water-bottle cork disassembly, it is also a disassembly application according to the previous classification.

4.1. Key disassembly from a lock

This application has to remove, disassembly, a key inserted in a lock. In this subsection the steps followed to perform the application are described according to the defined steps. In this application the tasks will be performed by one robot-arm, the PA-10 robot-arm with a parallel griper as tool.

The first thing to get after having the application is the system 3D model (Fig. 6). With this model a set of rules to perform the application are obtained. After getting the rules, there are to sort them to be done. The sequential list of rules is:

Rule 1: Grasp the key.

Rule 2: Remove the key from the lock.

Taking into consideration the tool that is required to perform the task, the two finger parallel griper, there are to compute the optimal grasping point. This point must be computed having into consideration the area of the key that is not accessible because it is inserted in the lock.

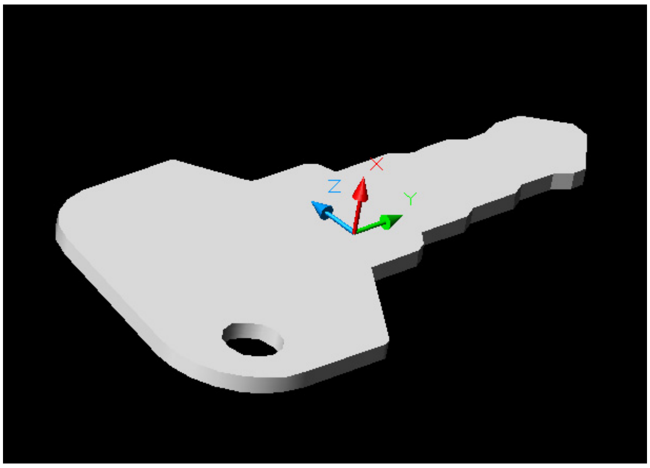

Fig. 6. 3D model of the key.

To compute the grasping point there are to choice the proper inertial 2D plane over the grasping point will be computed. The *xy* and *yz* planes have a very small surface to find an optimal grasping point; the plane *xz* is selected as the designed inertial plane. Once the manipulation plane is designed, the skeleton and the gravity centre, *c*, of the plane are computed. The gravity centre is computed by like the geometric centre of the blob that represent the object in the virtual image of the inertial plane.

With these information the point p_1 is computed from the skeleton according to (Díaz *et al.*, 2007) to hold the equation 3.

$$\min(\overline{p_1 c}) \qquad (3)$$

Taking into consideration the restrictions and the geometric model of the product, the grasping point obtained is shown in figure 7.

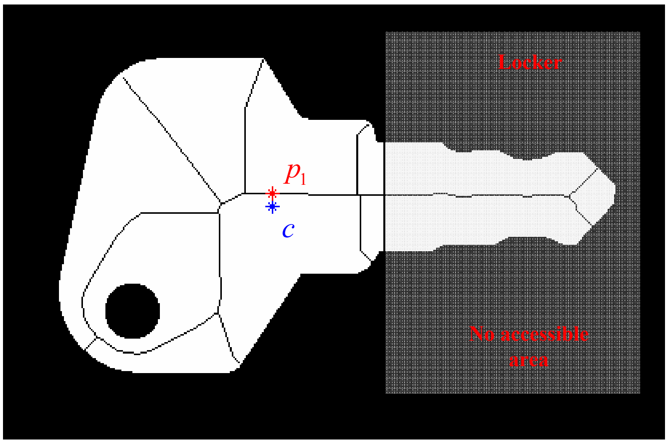

Fig. 7. Grasping point of the key assembled in the locker.

Once the grasping point is obtained, there are to compute the robot-arm trajectory from the current position to the grasping point. Furthermore, there are to compute the trajectory to accomplish the target objective defined by rule 2, the removal of the key trajectory.

The application of key disassembly from a lock performed by the robot arm is shown in the sequence of the Figure 8.

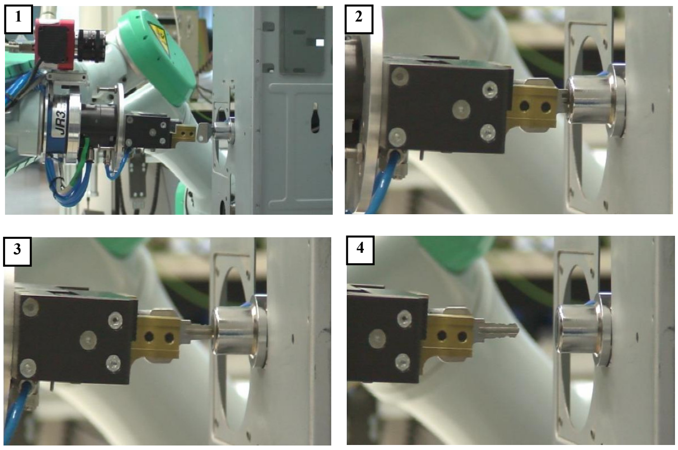

Fig. 8. Key disassembly sequence from a lock.

4.2. Water-bottle cork disassembly

This application has to disassembly the cork from a water-bottle. In this subsection the steps followed to perform the application are described according to the defined steps. In this application the tasks will be performed by one robot-arm, the Scorbot ER-IX robot-arm with a parallel griper as tool and by a human. This example includes cooperation human-robot to accomplish the target application.

After defining the application there are to have a model of the system, it is composed of the water-bottle model and the cork model. The model is shown in Figure 9.

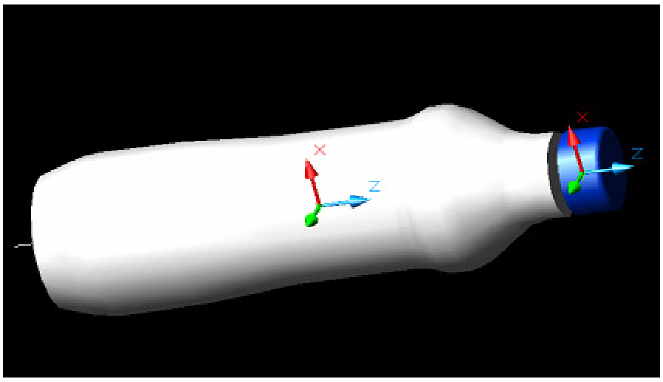

Fig. 9. Water-bottle model with cork.

Once the system is modelled there are to define the rules to be performed to open the bottle. The rules are:

Rule 1: Hold the bottle.

Rule 2: Catch the cork.

Rule 3: Remove the cork from the water-bottle.

This group of rules can be divided in a sequence of actions to be performed, ones by the robot and others by the human.

The first rule has the action of holding the bottle, it will be performed by the human, and the second rule, with the action of catching the cork will be performed by the robot, as well as the third rule.

Taking these rules and actions into consideration, the first and the second actions must be planed in parallel to compute an optimal grapping point for the robot and a reference of the grasping point that has to use the human.

Using the algorithm for compute the cooperative grasping, these points must hold the equation 4 to minimize the distance between the gravity centre of the product, c, and the weight distribution point, m, between the human grasping point h_1 and the robot grasping point p_1. Furthermore the algorithm tries to maximize the distance between the human and the robot to improve the security constrains during the task performance.

$$\begin{gathered}\min(\overline{mc}) \\ \max(\overline{p_1 h_1})\end{gathered} \tag{4}$$

The grasping points obtained are shown in Figure 10. The point h_1 is a reference for the human. It is not a strict grasping point because the system can not guarantee that the human will hold the bottle by this exact point. If the human hold the bottle by this point or by one near point is controlled by the supervision system. If the grasping point of the human is far away from the designed by the system, the robot grasping point must be recomputed taking the point h_1 as a know an fix point in the point where the human has hold the bottle.

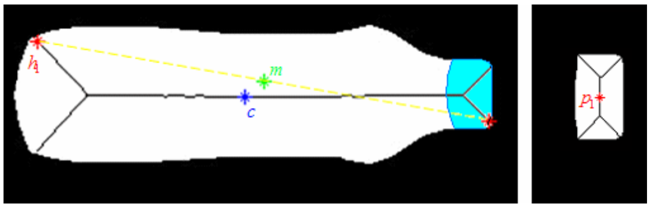

Fig. 10. Grasping point for the human, h_1, and the robot, p_1.

Once the grasping point are obtained the human hold the water-bottle and after the human hold the bottle the trajectory to catch the cork is computed from the current position of the robot to the grasping point compute previously. When it is computed the robot is move to catch the cork and then the algorithm to remove the cork is performed accomplishing the objective of the application.

A sequence of the application real application is shown in Figure 11.

5. CONCLUSIONS AND FUTURE WORK

A new uses of automatic disassembly systems has been show, with this approach a new field for this type of systems inside service robots is open.

The proposed systems glue together strategies for task planning, trajectory generation, grasping points; over a point of view of service applications.

In this field the future work can be using a great sensorization in the system to control its interaction with the environment. Another future work is the application of the proposed algorithms over humanoid robots.

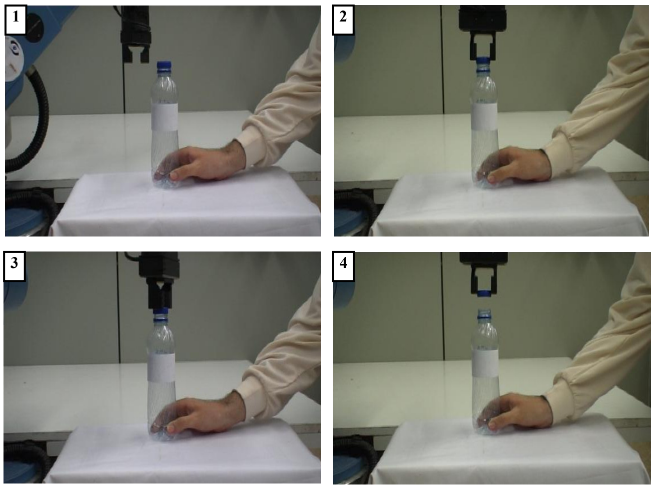

Fig. 11. Water-bottle cork disassembly sequence.

6. ACKNOWLEDGEMENTS

This work was funded by the Spanish MEC project "Diseño, implementación y experimentación de escenarios de manipulación inteligentes para aplicaciones de ensamblado y desensamblado automático" (DPI2005-06222).

REFERENCES

Díaz, C., S.T. Puente, F. Torres (2006). Tasks Planner of a Cooperative Robotic Dissasembly System. In: *12th IFAC Symposium on Information Control Problems in Manufacturing – INCOM-2006*. 163-168

Díaz, C., S.T. Puente, F. Torres (2007). Grasping points for handle objects in a cooperative disassembly system. In: *IFAC Workshop on Intelligent Assembly and Disassembly, IAD'07*, Alicante, Spain. 126-131.

Dougherty E.R., and R.A. Lotufo (2003). Hands Morphological Image Processing. In: *SPIE press: The International Society for Optical Engineering*. Bellingham, Washington. USA.

Fernández, C., O. Reinoso, M.A. Vicente and R. Aracil. (2006). Part Grasping for Automated Disassembly. *The International Journal of Advanced Manufacturing Technology*.**Volume 30 no 5**. 540-553.

Garcia, E., M.A. Jimenez, P.G. De Santos, M. Armada (2007). The evolution of robotics research - From industrial robotics to field and service robotics. *IEEE Robotics & Automation Magazine*. **Volume 14 no 1**. 90-103.

Gil, P., F. Torres, F.G. Ortiz, O. Reinoso (2006). Detection of partial occlusions of assembled components to simplify the disassembly tasks. *The International Journal of Advanced Manufacturing Technology*. **Volume 30 no 5**. 530-539.

Giudice, F., and G. Fargione (2007). Disassembly planning of mechanical systems for service and recovery: a genetic algorithms based approach. *Journal of Intelligent Manufacturing* **Volume 18 no 3.** 313-329.

Haralick R. and L. Shapiro (1992). *Computer and Robot Vision*. Addison-Wesley Publishing Company. **Volume 1 chap 5**.

Kang, J.G., and P. Xirouchakis (2006). Disassembly sequencing for maintenance: a survey. *In: Proceedings of the Institution of mechanical engineers part B-jorunal of engineering manufacture*. **Volume 220 no 10**. 1697-1716.

Kopacek, P. (2007) Semiautomatised disassembly of flat screens. In: *IFAC Workshop on Intelligent Assembly and Disassembly, IAD'07*, Alicante, Spain. 75-80.

Lambert, A.J.D. (2007). A heuristic for assembly and disassembly line balancing. In: *IFAC Workshop on Intelligent Assembly and Disassembly, IAD'07*, Alicante, Spain. 122-125.

Pomares, J., Torres, F. (2005). Movement-Flow-Based Visual Servoing and Force Control Fusion for Maniupulation Tasks in Unstructured environments. *IEEE Transactions on Systems, Man, and Cybernetics-Partc C*. **Volume 35 no 1**. 4-15.

Pomares, J., G. J. García, F. Torres. (2007) A Robust Approach to Control Robot Manipulators by Fusing Visual and Force Information. *Journal of Intelligent and Robotic Systems.* **Volume 48 no 4**. 437-456.

Serra, J. (2002), Course on Morphological Operators, CMM, École des Mines de Paris.

Soille P. (1999). *Morphological Image Analysis*. Springer.

Torres, F., S.T. Puente and R. Aracil (2003). Disassembly planning based on precedence relations among assemblies. *The International Journal of Advanced Manufacturing Technology*. **Volume 21 no 5**. 317-327.

Torres, F., and S.T. Puente (2006). Editorial: Intelligent disassembly in the demanufacturing process. *The International Journal of Advanced Manufacturing Technology*.**Volume 30 no 5**. 479-480.

Velásquez, J.D., and S. Nof (2007). Best-matching protocol for cooperation requirements planning in distributed assembly networks. In: *IFAC Workshop on Intelligent Assembly and Disassembly, IAD'07*, Alicante, Spain. 71-74.

Yu, J.C., and Y.M. Li (2006). Structure representation for concurrent analysis of product assembly and disassembly. *Expert Systems with Applications*. **Volume 31 no 4**. 705-714.