EURON/erf Technology Transfer Award

High Efficient Industrial Robotic Systems used for Palletizing and De-palletizing Operations for the non-flat Ceramic Industry

HTTP://ROBOTICS.DEM.UC.PT/NORBERTO/EURON_TTAWARD/

J. Norberto Pires
Mechanical Engineering Department
University of Coimbra
Portugal
1. Application Summary

Title: High Efficient Industrial Robotic Systems used for Palletizing and Depalletizing Operations for the non-flat Ceramic Industry.


Author Information: J. Norberto Pires, Mechanical Engineering Department, University of Coimbra, Polo II Campus, 3030 Coimbra, Portugal.
Email: norberto@robotics-dem.uc.pt
Web: http://robotics-dem.uc.pt/norberto/
Phone: +351.239.790712
GSM: +351.93.6864363
Fax: +351.239.790701

Company Information: Roca Cerâmica e Comércio SA, Leiria, Portugal

Contact Persons:
Engr. Manuel Neves, Head of Leiria Unit of Roca
Email: manuel_neves@roca.net

Engr. Sérgio Paulo, Industrial Systems
Email: sergio_paulo@roca.net

Very Important Note: Sérgio Paulo is co-author of this work, since part of it was developed under the program of his Master Thesis.

Important Note 1: The application presented here is composed by two systems, covering a wide collection of pieces and production requirements which poses very different problems and issues.

Important Note 2: Contact information of company contact persons is given for the jury only. Contacts are not public. Author requires special attention in handling those contacts. Public documents should not include contact persons details.

Application Website: This application and related multimedia material can be found in http://robotics-dem.uc.pt/norberto/euron_ttaward/
2. Motivation and Goals

This is a technical report about the development of a collection of prototype manufacturing cells, designed to perform automatic palletizing and de-palletizing operations of non-flat ceramic pieces such as toilets and bidets. The factories of these types of products show an impressive mixture of human and automatic labour, meaning that special attention must be taken with regard to human machine interfaces (HMI), safety, mode of operation, etc. In fact, today manufacturing setups rely increasingly on technology and it is really common to have all sources of equipment on the shop floor, commanded by industrial PCs or PLCs connected by an industrial network to other factory resources. Also, the production systems are becoming more and more autonomous, requiring less operator intervention in every day normal operation. That means using computers for controlling and supervision of the production systems, industrial networks and distributed software architectures. It means also designing application software that is really distributed in the shop floor, taking advantage of the flexibility provided by using programmable equipment.

Non-flat ceramic products are commonly used in our homes and are mainly associated with personal care tasks. The industrial production of these ceramic products poses several problems to industrial automation, especially if robots are to be used. Basically, these problems arise from the characteristics of the ceramic pieces: non-flat objects with high reflective surfaces, very difficult to grasp and handle due to the external configuration, very heavy and fragile, extensive surface sensitive to damage, high demand of quality on surface smoothness, etc. Also, the production setups for this type of products require very high quality and low cycle times, since this is a large scale industry that will only remain competitive if production rates are kept high. Another restriction is related to the fact that this industry changes products very frequently, due to fashion tendencies in home decoration, etc. Also, there is the already mentioned mixture of automatic and human labour production, which is a difficult problem since HMI are very demanding and a key issue in modern industrial automation systems.

It was proposed to us by the partner company to build several de-palletizing and palletizing solutions, with very simple and graphical operator interface, to install in their final inspection lines. In those lines human operators inspect all pieces by hand to find functional and surface defects (computer vision solutions for inspection). The challenge was to build high efficient systems, capable of handling more pieces a day than its human counterparts, that could be very easy to setup.
and startup at the beginning of the day. So, there is a robotic challenge and a software challenge, namely, in designing human-machine interfaces for operators.

**Specific Goals**

The main objective was to build a robotic system (Fig.1) that could be used to feed the final inspection lines, working from pallets (input-pallets) coming from high temperature oven, remove then after inspection and assemble the final pallets (output-pallets). The system should be able to work with input-pallets composed by 4 levels of ceramic pieces, 8 pieces per level placed in a special order to keep pallet equilibrium, and with levels separated with pieces of hard paper. It should also be able to work with output-pallets up to 5 levels of ceramic pieces, 8 pieces per level placed in the same order as in the input-pallets, with levels also separated by hard paper. The rule used to arrange the pieces in the pallet is to place them alternatively one up – one down, starting from the ground level, then swap to one down – one up in the next level (Fig. 2) and keep the procedure in the proceeding levels.

**Figure 1** – Components of the system.

Actually input-pallets are assembled manually by operators at the end of the high temperature oven. This means that the robotic system must be tolerant with possible medium-large palletizing errors, coming from misplaced pieces both in position and orientation, and showing also significant variations from level to level. Another important thing is that pallets are fed into the system by human operators using electro-mechanic pile drivers, which also introduces some variation in the
pallet. Sometime in the future AGVs will be use to fulfill the task, reducing considerably the variations introduced and increasing efficiency of the system.

View of the system.

**Figure 2** – Pallets and view of the system.
Other important requirements include:

1. Possibility to easily introduce new product models;
2. Possibility to parameterize the operation on each model, so that best performance per model is achieved;
3. Possibility to change model under production without stopping production;
4. Possibility to configure output-pallets according to client requirements. Clients require output-pallets ranging from 3 levels (16 pieces) up to 5 levels (40 pieces).
5. Possibility to monitor production using a graphical interface;
6. Obtain medium cycle times of about 12 seconds per piece, both at feeding the inspection lines and at removing the accepted pieces for the final pallets (output pallets).

**Approach and Results**

Considering the objectives and requirements of this project, it was decided to design a robotic cell that could handle the ceramic pieces under consideration. Proper grippers and layouts were designed and built. The robots work as slaves to a central PC, where all the parameterization is performed. The PC also monitors the operation, being of guidance when something wrong happens. Operator is able to solve problems from the PC. There is one PC for each robot. That was done for practical and simplicity reasons, but it is not a requirement.

Having these objectives in mind it was decided to operate the system using an external personal computer, using the teach pendant of the robot only for a few special routines not necessary in every day normal operations. Client-server software architecture was adopted. The robot controller software works as a server, exposing to the client a collection of services that constitute its basic functionality. A collection of services was designed to fulfill all the tasks required to the system, so that they could be called from the PC (Fig. 3). The software architecture used in this work, was presented in detail elsewhere [2,3], and is distributed using a client-server model, based on software components developed to handle equipment functionality.

The system is completely operated using a graphical panel running on the PC, built using the above mentioned ActiveX controls in Visual C++. When the system is started, the operator needs only to specify what product model will be used in each pallet, and if first pallets are fully assembled. This need is only for the de-palletizing
sub-system, because there is no identification on the pieces (they are coming from the high temperature oven). On the palletizing sub-system there is no need to specify the model, since the pieces carry barcodes, inserted by the inspecting operator, that are used by the sub-system, with the help of barcode readers, to identify the model.

![Software architecture used.](image)

**Figure 3** – Software architecture used.

Sometimes, due to production there are some non-fully assembled input-pallets on the shop floor, and there is the need to introduce those pallets in the system. To be able to do that, the software allows operator to specify the position and level of the first piece. That is however only possible on the first pallet, because the system resets definitions to the next pallets to avoid accidents, i.e., proceeding pallets are assumed to be fully assembled. The same happens with output-pallets, since the system must be able to fill a pallet not completely filled on the last production cycle for that model.

**Basic functioning for de-palletizing system:** When the operator commands “automatic mode” the robot approaches the selected input-pallet in direction to the actual piece, searches the piece border using optical sensors placed on the gripper, and fetches the ceramic piece. After that the robot places the piece in the first available inspection line, alternating inspection lines if they are both available, i.e.,
the robot tries to alternate between them, but if the selected one is not available then the other is used if available. If both inspection lines are occupied the robot waits for the first to became available. Figure 4 shows the interface used by the operator to command the system and monitor production. It shows the commands available, and the on-line production data that enables operators to follow production. All commands and events are logged into a log file, so that production managers can use it for production monitoring, planning, debugging, etc. The system uses also a database, organized in function of the model number, where all the data related to each model is stored. That data includes type of the piece, dimensions, height where the gripper should grab the piece, average position of the first piece of the pallet, height of the pallet, etc. Accessing and updating the database is done in “manual mode”, selected in the PC interface. There is a “teaching” option that enables operator to introduce new models and parameterize the database for that model. When that option is commanded, the robot pre-positions near the input-pallet and the operator can jog the robot using function keys to the desired position/orientation. Basically the de-palletizing operation is preformed step-by-step and the necessary parameters acquired in the process, inquiring the operator to correct and acknowledge when necessary. The operator is only asked to enter the “model number” to teach, the height and the width of the piece. The rest is automatic. After finishing this routine the model is introduce into the database, and the system can then work with that model number.

System is able to check for errors like: wrong pallet for model, presence of pallet, model not known, no piece in place, wrong level, etc. Proper warnings and sent to the PC for operator information.

**Basic functioning for palletizing system:** A similar approach was used. Two lines are also used, with the robot trying to alternate between them. But the first available piece is removed not slowing down production. Similar approach to the one used with the de-palletizing sub-system is use to “teach” models to the robot. Also the system identifies the model number from the piece barcode, when “automatic-mode” is commanded, fetches the piece and inserts it in the pallet compatible for that model. Operator is able to select what pallet to use first, how many pieces already there, and how many pieces it should carry. Do to the required dimensions of the output-pallets, the robot was placed on the top of a linear-axis, controlled by the robot control system (robot external axis), so that a wider area could be reached. System is also able to check for errors like: wrong pallet for model, presence of pallet, model not known, no piece in place, etc. Proper warnings and sent to the PC for operator information.
Figure 4 – Example of interface used by operators (de-palletizing system)

Figure 5 – Example of interface used by operators (palletizing system)
Operational Results
The system achieved the required operational results and is flexible enough when considering introduction of new models. Currently it works 2 shifts a day, almost autonomously, making around 1800 pieces per shift (one shift is of about 7.30 hours). Operators adapted very easily to the system, since they only interface with a touch-screen very easy to use.
Company improved quality of production and reduced production costs: less operators are needed and production is more efficient (more pieces a day). This can be demonstrated by operational results, and also by the fact that new systems followed this one to handle other type of pieces and other type of operations, resulting in a strong connection between university and this company.

Innovations and Technology Transfer
The following innovations and technology transfers were successfully introduced with this project (and others that followed this one):

1. A Human-machine interface for robotic cells was introduced with very good results. The interface was done using Visual C++ and works very well.
2. A client-server model in programming robot cells, integrating all the cell functionalities was introduced and installed. Operation and parameterization is completely done on the PC. The model is based on Remote Procedure Calls issued to server programs installed on the client computer systems: robot controller, for example. The programming PC has also the capability of receiving event calls from the client computers, implement in this way an efficient warning mechanism.
3. A very efficient system was obtained using commercial robots and systems (like PLCs, sensors, etc.) connected to the central PC over an Ethernet network. No commercial software was used in this project, besides the operating systems (Baseware 4.0 for robots and Windows 2000/SP4 for PCs) and developing tools (Visual C++ 6.0). A port of SUN RPC 4.0 to Windows 2000 was also used, but the porting effort was completely done by the author.
References


Videos and pictures

Videos and pictures can be found in:

http://robotics.dem.uc.pt/norberto/euron_ttaward/

Link to RPC software used: http://robotics.dem.uc.pt/norberto/pprob5/