Available online at www.sciencedirect.com

SCIENCE DIRECT®

Robotics and Computer-Integrated Manufacturing 21 (2005) 11–17

Robotics and Computer-Integrated Manufacturing

www.elsevier.com/locate/rcim

Complete robotic inspection line using PC-based control, supervision and parameterization software

J. Norberto Pires[a,]*, Sérgio Paulo[b]

[a] *Mechanical Engineering Department, University of Coimbra, Coimbra, Portugal*
[b] *Roca Comércio e Cerâmica SA, Leiria, Portugal*

Abstract

Non-flat ceramic products, like toilets and bidets, are fully inspected at the end of the production process, to search for structural, surface and functional defects. Ceramic pieces are transported to the inspection lines assembled in pallets, carried by electro-mechanical fork-lifters or automatic guided vehicles. Pallets need to be disassembled, while feeding with the inspection lines where human operators execute the inspection tasks. Also, the pieces that pass inspection need to be palletized again for product distribution. Those de-palletizing and palletizing operations are physically very demanding and can be performed with advantages by robots.

This paper discusses with enough detail the adopted solutions used to perform those tasks, giving special attention to the software designed to parameterize and supervise the system.

Keywords: Industrial robotics; Distributed software; Palletizing and de-palletizing

1. Introduction

This paper is a case study reporting 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 [1]. Also, the production systems are becoming more and more autonomous, requiring less operator intervention in day to day normal operation. That means using computers for controlling and supervision of the production systems, industrial networks [2] and distributed software architectures. It also means, designing application software that is really distributed in the shop floor, taking advantage of the flexibility provided by using programmable equipment. Consequently, there is an extensive mixture between human and machine labour in the actual manufacturing systems. The situation, although desirable since agility is a key issue for productivity and efficiency of actual companies, poses some difficulties to the efficient cooperation between humans and machines. Human–machine interfaces (HMI) are consequently very essential to take advantage of the flexibility and capabilities of installed production machines, in cooperation with the human operators [1].

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, difficulty to grasp and handle due to the external configuration, heavy weight and fragility, extensive surface sensitivity to damage, high demand on quality of surface smoothness, etc. Also, the production set-ups for this type of products require high-quality and

*Corresponding author.

0736-5845/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.rcim.2004.05.003

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 product designs 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 simple and graphical operator interface, to install in their final inspection lines. On these lines, human operators inspect all pieces by hand to find functional and surface defects (computer vision solutions for inspection). The challenge was to build highly efficient systems, capable of handling more pieces a day than its human counterparts, that could be 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 the operators.

The system presented here was designed to take advantage of computers and available tools to parameterize and monitor an industrial robotic cell, i.e., to make human–machine interface. In the process of describing and discussing the system a few available technical details are highlighted. This is also important due to the fact that the entire software was built from scratch [3], without using any of the available commercial software packages.

The paper is organized as follows: Section 2, which follows this introduction, presents the industrial problem addressed in this paper, the scientific and technical challenges it places to the developing team, along with a clear definition of the objectives of the described industrial project. Section 3 describes the solution adopted for this project, focusing mainly on the software developed to handle the human–machine interface. Section 4 discusses the results obtained, highlighting the technical innovations and the technical transfers to the industry introduced by implementing the solution designed to meet the objectives and challenges of this industrial project.

2. Motivation and goals

The problem addressed in this paper is the construction of a complete system to assist humans in the task of inspecting non-flat ceramic pieces. Those pieces (bidets and toilets, mainly) organized in pallets (input-pallets), reach the inspecting site directly from the high-temperature oven, using fork-lifters. A few operators placed along two inspecting lines (15 m long each), inspect all the pieces by hand, searching for pieces with functional and surface defects, and removing from the inspection lines the rejected pieces [4]. Consequently, in this system there is the need to de-palletize the input-pallets, continuously feeding the two inspection lines. The system must also be able to pick the accepted pieces from the end of the inspection line, palletizing them again into the pallets (output-pallets) used for product distribution (Fig. 1). The system should work also as autonomously as possible, requiring only minor parameterization at the beginning of the work or the production cycle. 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 the levels separated by pieces of hard paper. It should also be able to work with output-pallets upto 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.

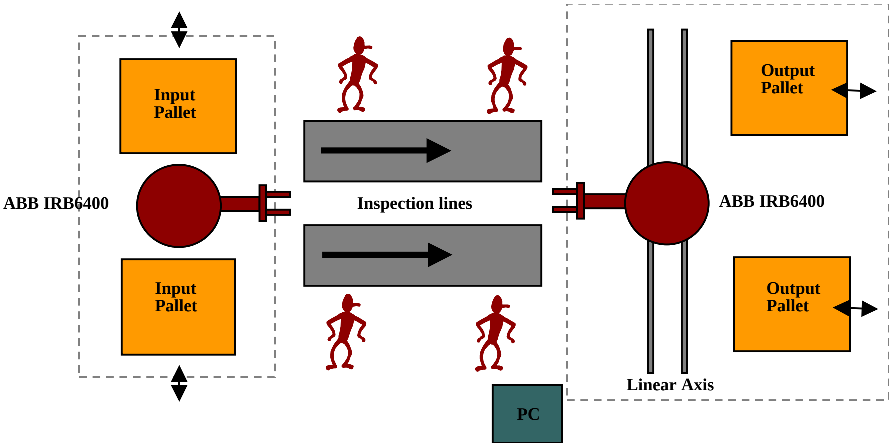


Fig. 1. Components of the system.

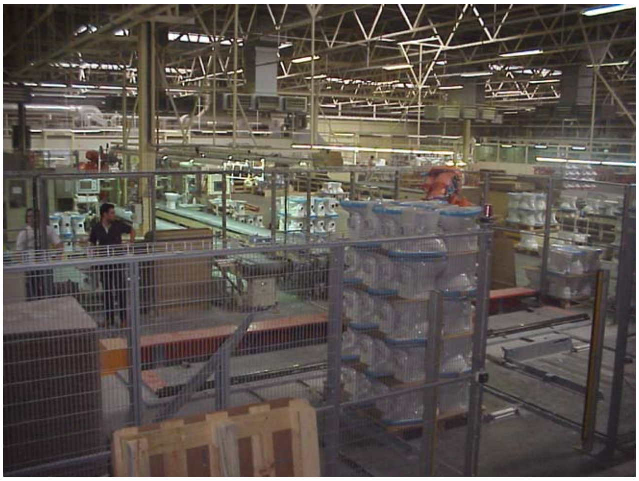


Fig. 2. Pallets and view of the system.

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, originating from misplaced pieces both in position and orientation, and also showing significant variations from level to level. Another important thing is that the pallets are fed into the system by human operators using electro-mechanic fork-lifters, which also introduces some variation in the pallets. Sometime in the future Automatic Guided Vehicles (AGV) will be used to fulfill the task, reducing considerably the variations introduced, and increasing the efficiency of the system.

The main objectives for this system are summarized as follows:

1. Build a complete robotic system capable of performing de-palletizing and palletizing operations to assist inspection lines.
2. The system must perform each of these operations in less than 12 sec per piece.
3. The system should cope with high palletizing errors on the input-pallets, since they are assembled by human operators which permits to anticipate small–medium placement errors (up to 5 cm in position and up to 5° around the vertical axis).
4. The system should cope with deviations in pieces dimensions up to ± 1 cm in each dimension. Ceramic pieces grow inside the high-temperature oven, causing these deviations expected due to temperature deficiencies, variation of time inside the oven, variations in the ceramic mixture, etc. These deviations are not necessarily errors, but instead a characteristic of this type of production.
5. The system must work with both input and output pallets, with variable number of pieces, comprising of any number of pieces, in the case of the input pallets, and 8, 16, 24 or 40 pieces in the case of the output pallets.
6. The system should maintain information about its surroundings, in a way to warn about inconsistencies between what is ordered and what is available.
7. The system must be parameterized easily, using a graphical interface implemented to be used with a touch-screen. A few commercial software packages are available in the market. Nevertheless, our option was to build our own solution since the human–machine interface plays a crucial role in the performance of the system, including operator acceptance. The authors therefore think that it is very important to have full control of the developed software.
8. The system must be optimized for each model of ceramic pieces. This means that there should be the possibility to introduce new models using a "teach" strategy.

Considering the above-mentioned objectives the following challenges were identified:

1. To build a human–machine interface, easy to use and capable of handling production needs and operator interface. System warnings and errors must be brought to the operator's attention in an efficient way. All operations and messages must be logged for future analysis.
2. To build a system capable of meeting the planned requirements.
3. To explore the capabilities of the current personal computers, operating systems and related tools on a very demanding industrial environment.

Taking the above-defined objectives and challenges, and considering the fact that this is an industrial project, meaning that it is supposed to work 24 h a day without problems, it was decided to distribute the software to all the components of the system. A client–server architecture [5–8], based on remote procedure calls (RPC) [9], was adopted, with the PC being the client of the rest of the components of the system, including the robot controllers, having also the already mentioned mission of interfacing with operators.

3. 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. It was also 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. The robots work as slaves to that central

PC, where all the parameterization is performed. The PC also monitors the operation, being of guidance when something goes wrong. The operator is able to solve problems from the PC. There is one PC for each robot. This was done for practical and simplicity reasons, but it is not a requirement.

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 by 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 [3,5], and is distributed using a client–server model, based on software components (ActiveX controls) [10,11] 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++. NET 2003 [12]. When the system is started, the operator needs only to specify what product model will be used in each pallet, and if the 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.

Sometimes, 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 the 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 an incompletely filled pallet on the last production cycle for that model.

3.1. 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

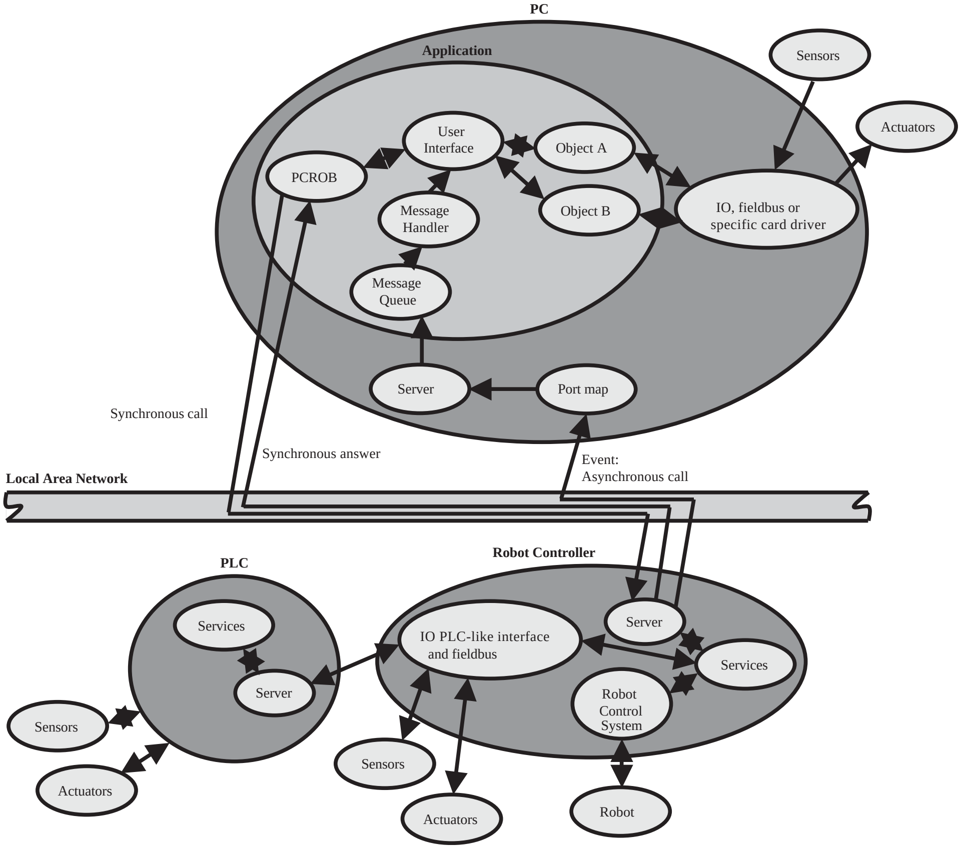
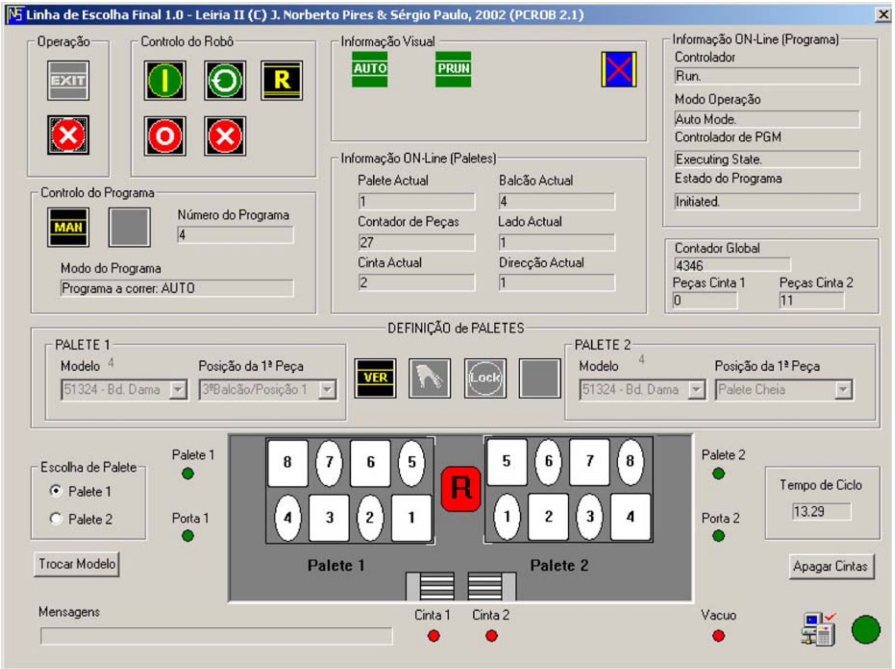


Fig. 3. Software architecture used (PCROB is an ActiveX control for robot control).

then the other is used if available. If both inspection lines are occupied the robot waits for the first to become available. Figs. 4 and 5 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 also uses a database, organized in function of the model number, where all the data related to each model are stored. That data include type of the piece, dimensions, height where the gripper should grab the piece, average position of the first piece of the pallet,



Linha de Escolha Final 1.0 - Leiria II (C) J. Norberto Pires & Sérgio Paulo, 2002 (PCROB 2.1)

Informação ON-Line (Programa)
Controlador: Run.
Modo Operação: Auto Mode.
Controlador de PGM: Executing State.
Estado do Programa: Initiated.

Informação ON-Line (Paletes)
Palete Actual: 1
Balcão Actual: 4
Contador de Peças: 27
Lado Actual: 1
Cinta Actual: 2
Direcção Actual: 1

Controlo do Programa
Número do Programa: 4
Modo do Programa: Programa a correr: AUTO

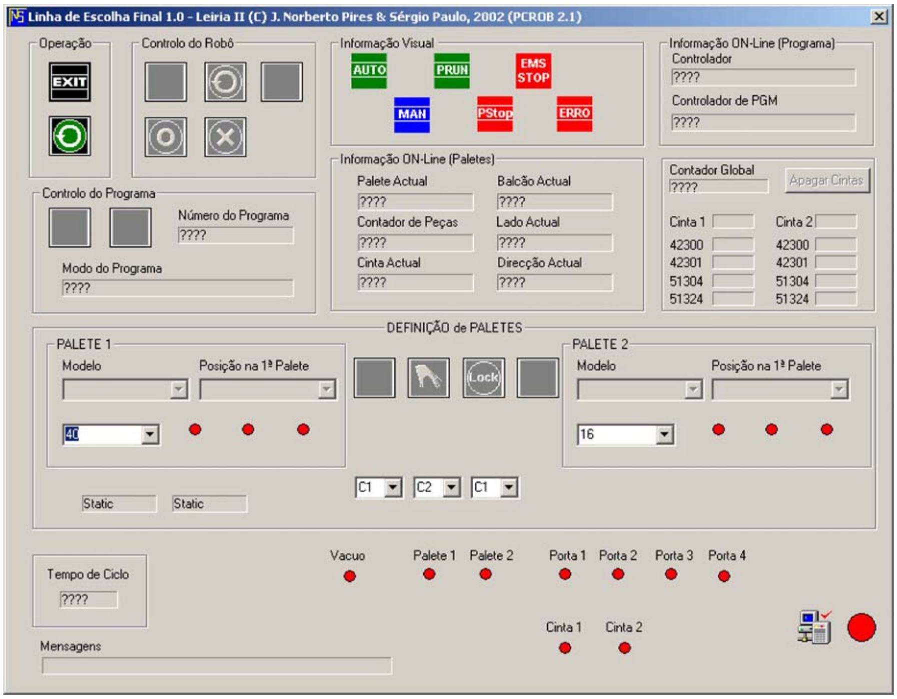
Contador Global: 4346
Peças Cinta 1: 0
Peças Cinta 2: 11

DEFINIÇÃO de PALETES
PALETE 1 — Modelo 4: 51324 - Bd. Dama; Posição da 1ª Peça: 3ºBalcão/Posição 1
PALETE 2 — Modelo 4: 51324 - Bd. Dama; Posição da 1ª Peça: Palete Cheia

Escolha de Palete
◉ Palete 1
○ Palete 2

Tempo de Ciclo: 13.29

Fig. 4. Example of interface used by operators (de-palletizing system).



Linha de Escolha Final 1.0 - Leiria II (C) J. Norberto Pires & Sérgio Paulo, 2002 (PCROB 2.1)

Informação ON-Line (Programa)
Controlador: ????
Controlador de PGM: ????

Informação ON-Line (Paletes)
Palete Actual: ????
Balcão Actual: ????
Contador de Peças: ????
Lado Actual: ????
Cinta Actual: ????
Direcção Actual: ????

Controlo do Programa
Número do Programa: ????
Modo do Programa: ????

Contador Global: ????

Cinta 1		Cinta 2	
42300		42300	
42301		42301	
51304		51304	
51324		51324	

DEFINIÇÃO de PALETES
PALETE 1 — Modelo; Posição na 1ª Palete; 40
PALETE 2 — Modelo; Posição na 1ª Palete; 16
C1 C2 C1
Static Static

Tempo de Ciclo: ????

Fig. 5. Example of interface used by operators (palletizing system).

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 the operator to introduce new models and parameterize the database for that model, where a "teach by showing" strategy is used. 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 performed 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 introduced into the database, and the system can then work with that model number.

The 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 are sent to the PC for operator information, and displayed using software icons and/or short messages.

3.2. Basic functioning for palletizing system

A similar approach was used. Two inspection lines are also used, with the robot trying to alternate between them. But the first available piece is removed without slowing down the production. A similar approach to the one used with the de-palletizing sub-system is used 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. The operator is able to select what pallet to use first, how many pieces are already there, and how many pieces it should carry. Due 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. The 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 are sent to the PC for operator information, and displayed using software icons and/or short messages.

4. Operational results and discussion

The system achieved the required operational results and is flexible enough when considering introduction of new models. Currently it works two shifts a day, almost autonomously, making around 1400 pieces per shift (one shift is of about 7 h 30 min). Operators adapted easily to the system, since they only interface with a touch-screen which is easy to use.

The 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 the university and this company.

A few innovations and technology transfers were successfully introduced with this project and the others ongoing with the same company [3,6]. *A fairly interesting human–machine interface* for robotic manufacturing cells was introduced showing good results [3,6]. The solution has been developed from the scratch using Visual C + +. NET 2003, constituting a software platform that can be used with other applications. Experience with operators is very positive, showing that they have adapted well and really enjoy using it. Nevertheless, new developments are necessary in a way to guide operators and reduce operator training. This means that advanced help should be available to guide the operator, when inconsistencies are detected. Such inconsistencies include, for example:

1. Commanding "automatic mode" without reviewing the pallets parameterization. That could be correct in some situations and because of that allowed. At the moment only a visible warning is issued, but in the future only some sequence of operations will allow "automatic mode".
2. Ordering a "RSTART", i.e., proceed with current configuration and from the same program position, after a system stop due to an error or operator manual stop. Actually this situation is permitted, after confirmation with password, since we still rely on operator training and judgment. Nevertheless, in future, operators should be guided to follow a certain procedure, reviewing actual status, in a way to avoid mistakes. This can certainly be done, for example, using an inference mechanism based on fuzzy-logic.

The two presented situations are good examples of the future developments that are needed. For a certain industrial robotic cell characterized by a set of available operations, a collection of routes should be defined considering all possible operational situations. Consequently, an operator can command the robotic cell if he follows one of those routes or procedure. This will increase safety, avoid errors and improve efficiency. At the moment, critical operations require operator confirmation using user password, with visible warnings issued to the screen.

Another interesting innovation was the utilization of a client–server architecture, explained elsewhere [3,5], developed by the first author to be used with robotic cells. Using this architecture implies the clear intention to distribute functions to all "intelligent" components of the robotic cell, leaving to the central PC (the client) the tasks of making the service request calls, properly parameterized, and displaying system information to the user. The PC is the user commanding interface, and his window to the system.

5. Conclusions

In this technical paper, a robotic system designed to assist inspection operations in the non-flat ceramic industry is presented and discussed. The system involves de-palletizing and palletizing operations of different pieces, requiring minor information from the operators. A human–machine interface was designed to facilitate user operation, and showed good operational results. A few improvements to the human–machine interface software platform were identified and are being implemented in our laboratory. Finally, it is worthwhile to stress that the entire software was developed from the scratch using available operating systems and selected software tools, without using any commercial software package in any of the tasks. This paper may also be useful because of that reason.

Acknowledgements

The authors wish to express their gratitude to ABB Robotics Portugal, especially to Mr. Paulo Monteiro, for the cooperation in this project, namely providing access to detailed technical information of the new ABB S4 C+ robot controller.

References

[1] Kusiak A. Computational intelligence in design and manufacturing. New York: Wiley; 2000.

[2] Halsall F. Data communications, computer networks and open systems, 3rd ed. Reading, MA: Addison-Wesley; 1992.

[3] Pires JN, Sá da Costa JMG. Object-oriented and distributed approach for programming robotic manufacturing cells. IFAC J Robot Comput Integrated Manuf 2000;16(1):29–42.

[4] Pires JN, Paulo S. High-efficient de-palletizing system for the non-flat ceramic industry. Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei; 2003.

[5] Pires JN. Object-oriented and distributed programming of industrial robotic and automation equipment. Ind Robot, Int J 2000; 27(4).

[6] Pires JN. Using Matlab to interface industrial robotic and automation equipment. IEEE Robot Automat Mag 2000; 7(3).

[7] Pires JN, Ramming RS, Araújo R. Force/torque sensing applied to industrial robotic deburring. Sensor Rev J, Int J 2002;22(3): 232–41.

[8] Pires JN, Godinho T, Ferreira P. CAD interface for automatic robot welding programming. Ind Robot, Int J 2004; 31(1).

[9] Bloomer J. Power programming with RPC. Sebastopal, CA: O'Reilly & Associates, Inc.; 1992.

[10] Box D. Essential COM. Reading, MA: Addison-Wesley; 1998.

[11] Rogerson D. Inside COM. Microsoft Press; 1997.

[12] Visual C++. NET 2003 Programmers Reference, Microsoft; 2003 (reference can be found at Microsoft's web site in the Visual C++. NET location).