

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



The robot voice-control system with interactive learning

Miroslav Holada, Martin Pelc
*Technical University of Liberec
Czech Republic*

1. Introduction

Nowadays, robots are penetrating human lives and carry out many tasks that would have been impossible for machines just few decades ago. One can usually get an “end user” narrow- or single-purpose robotic solution for a specific task with a user interface providing control within the scope of the given task. Another possibility is to get a more general-purpose robot and design a custom control system for peculiar tasks. The latter usually involves programming using the manufacturer's rudimentary robot movement interface and requires deeper knowledge of robotics, computer sense, control systems etc.

Providing a general-purpose control system for general-purpose robots which wouldn't require most of the aforementioned knowledge poses a challenge. It would make robotics appealing to wider audience and possibly speed up development and application of robots. Such system needs to have some “learning” capabilities as well as a friendly user interface for task “teaching”.

The goal of our work described in this capture is a PC-software-based interactive system for general-purpose robot voice-control. This paper describes the designed prototype, its structure and the dialogue strategy in particular. The interactive control of robots could be used in special situations, when a robot is working in dangerous areas and no programming beforehand is possible. It could also be used in a situation when supervised learning for robot's later autonomous operation has to be done, without knowledge about the robot's programming language.

Generally, the robots are actuated by sets of control commands, sometimes by a manual control interface (such as touchpad or joystick). The operator has to know the control commands, syntax rules and other properties necessary for successful robot control. The proposed system tries to simplify this robot programming and make it more user-friendly and easy to use. The system offers commands like “move left” or “elevate arm” that are translated and sent into the corresponding device (robot).

2. Project features

The project is based on a former research. The research involved a voice-control dialog system, speech recognition, vocabulary design and speech synthesis feedback for user

command confirmation. Together with a scene manager and a digital image processing module, it forms the core of the control system as shown in figure 1.

The key feature of the system is that it can “learn” a series of commands to autonomously perform certain tasks using the robot. Digital cameras will be used to navigate in the robot's working space. Supported by computer vision algorithms, the system should be able to find objects of interest (and to keep track of them) on the scene including the robot itself and allow the objects to be referred to by user's commands. The main feature is that a user is not required to have knowledge about robot programming, computer vision, etc.

The system should also offer a straightforward robot movement control via either verbal commands or graphical user interface (GUI).

The system is being developed as a pure-software solution hosted on the Windows platform. The components of the system are described below.

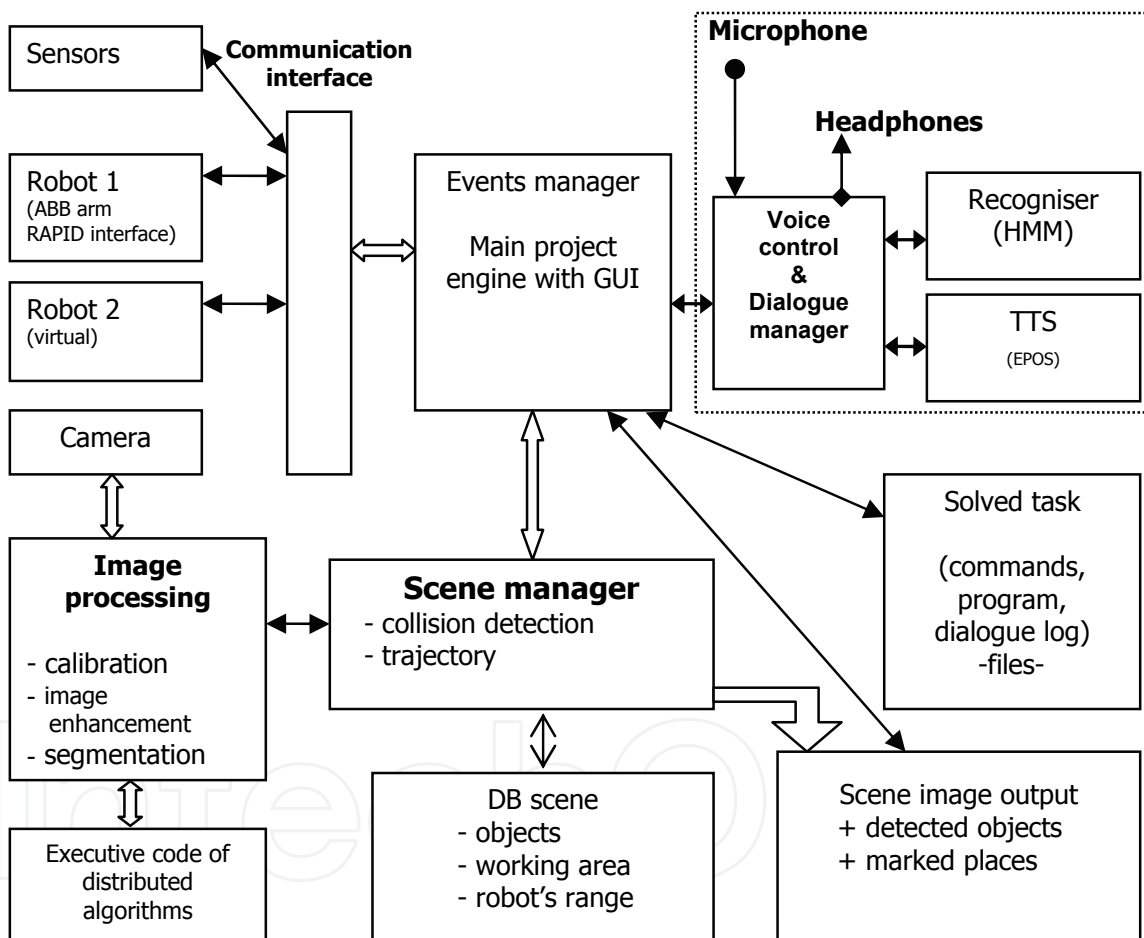


Fig. 1. Functional layout of the system.

2.1 Scene manager

The scene manager forms a connection between the main program (engine) and the image processing part. It actually controls the image processing module and initiates image acquisition and processing. Using the processed image data, it updates the scene database, keeps track of objects found on the scene and provides the scene object and image data to

the main engine. It is also aware of the robot's coordinate system and plans the robot's movement when requested by the engine.

The database itself consists of two types of data. It contains the list of parametrized objects detected on the scene as well as the robot calibration data. The latter allows mutual image-space to robot-space coordinate translation which is used in robot navigation. Each object detected on the scene is internally represented as a data object (class instance), all the objects are stored in a dynamic list. Some of the attributes are: a unique object identifier, object's shape descriptor, central point coordinates, bounding rectangle etc. Such data allows smooth object manipulation and serves as a base for object collision avoidance along the manipulation trajectory.

The scene manager also combines unprocessed camera image with scene data to highlight detected objects and to present them to the user via a GUI as shown in figure 2. The user has a view of the computer's scene understanding and may correctly designate objects of interest in his or her commands.

Being in its early stages, the project currently works only with 2D data and relies on the user's z-axis navigation aid. The system is expected to incorporate a second camera and 3D computer vision in the future to become fully 3D aware.

2.2 Speech processing

The voice interface between an operator and the controlled process is provided by a speech recogniser and a text-to-speech synthesis (TTS) system (both for Czech language). The TTS synthesis system named EPOS was developed by URE AV Prague. It allows various male or female voices with many options of setting (Hanika & Horak, 1999).

The speech recognition is based on a proprietary isolated word engine that was developed in previous projects (Nouza, 2000). The recogniser is speaker independent, noise robust, phoneme based with 3-state HMM (Hidden Markov Models) and 32 Gaussians. It is suitable for large vocabularies (up 10k words or short phrases) and allows us to apply various commands and their synonyms (Nouza & Nouza, 2004).

Both voice components are built into a distributed system named DUNDIS (Holada, 2004). The advantage of this solution is the fact that the designed system needs to incorporate only a relatively simple software client. This client sends speech data to the recognition server where speech recognition is executed. The TTS engine can work separately but in this case it is linked to recognizer due to echo cancellation problem. The unwanted acoustic feedback (meaning that the computer "hears" and recognizes what it "speaks") is eliminated by half-duplexing the communication (it either "speaks" or "listens" but not both at the same time).

2.3 Image processing

The robot's working area is captured by a colour high-resolution digital camera (AVT Marlin F-146C, 1/2" CCD sensor). The camera is placed directly above the scene in a fixed position. We implemented a simple interactive method to synchronize the robot's coordinate system (XY) and the camera's one using pixel units and prepare modifications to compensate geometric distortions introduced by camera lens.

The picture 2 shows the overall view of the test workplace. The camera is placed above the scene and is partially visible on the top of the picture. The working scene is composed of the

robot's surrounding, most notably the white desk with disks that are placed on ribbons to prevent robot's tool damage (crashing directly into the desk).

Digital image processing methods are placed in a library which is served by the scene manager with the object database. The figure 3 shows the circular object detection using the reliable Hough transform (HT). HT is commonly used for line or circle detection but could be extended to identify positions of arbitrary parametrizable shapes. Such edge-based object detection is not too sensitive to imperfect input data or noise. Using a touch-display or verbal commands it is possible to focus the robot onto a chosen object (differentiated by its color or numbering) and then tell the robot what to do.

So far the system supports detection of basic geometric shapes (circle, rectangle) and basic colors.

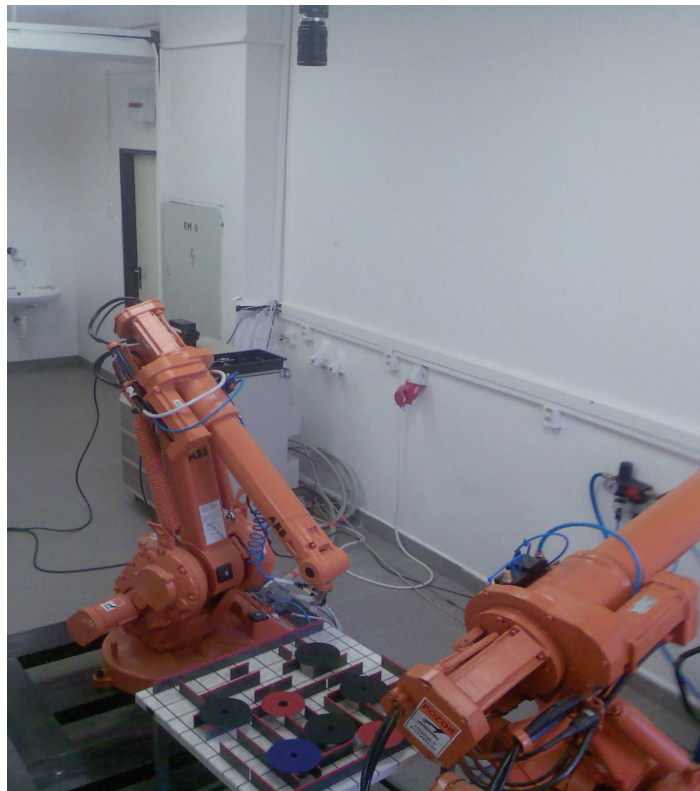


Fig. 2. Overall view of the test workplace with robot, working scene and the camera.

2.4 Robots description

For the purpose of debugging the system, a virtual robot device was designed which behaved like the real one but worked only as a graphical computer simulation. For field tests a real robot had to be used and we chose an industrial robot typically used in robotics lessons which was available to us.

The prototype system uses a compact industrial general-purpose robotic arm (ABB IRB 140). The robot is a 6-axes machine with fast acceleration, wide working area and high payload. It is driven by a high performance industrial motion control unit (S4Cplus) which employs the RAPID programming language. The control unit offers extensive communication capabilities - FieldBus, two Ethernet channels and two RS-232 channels. The serial channel

was chosen for communication between the robot and the developed control system running on a PC.

The robotic control software module simplifies the robot use from the main engine's point of view. It abstracts from the aspects of physical communication and robot's programming interface. It either accepts or refuses movement commands issued by the core engine (depending on command's feasibility). When a command is accepted, it is carried out asynchronously, only notifying the engine once the command is completed.

Industrial robots have their own sophisticated control systems which allow arbitrary task programming. In our case, the RAPID-based ABB control system proved to be not very suitable for applications which require direct movement control by the computer program, not just the RAPID program stored in the control system.

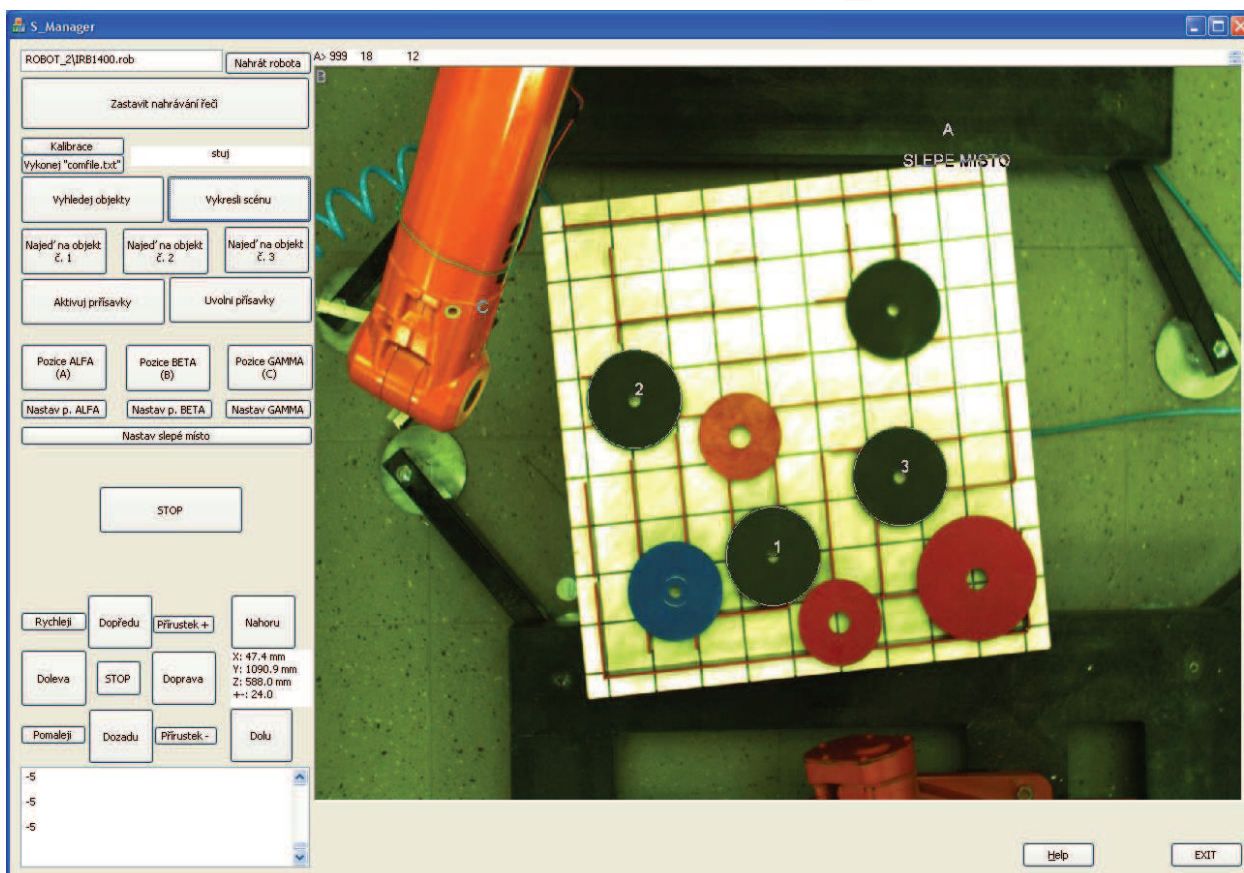


Fig. 3. The system's GUI with the scene view and highlighted objects.

2.5 Distributed computing

Most of the system's modules are developed and run on a standard PC to which the robot is connected. Since some of the software modules require significant computational power, the system's response time was far from satisfactory when the whole system ran on a single computer. Therefore, the most demanding computations (namely the object recognition and the voice recognition) were distributed to other (high performance) computers via network (TCP connections).

The solution with distributed components is advantageous especially for research and debugging. If any part of the system crashes then after its restart the other parts are quickly

reconnected without need to reload and initialize them. Today's local networks are fast enough so any introduced transfer delays are insignificant.

3. Dialogue strategy

The dialogue scenario contains four vocabularies. The first is composed of simple basic control commands like "move up", "stop" or "take it". They are necessary for basic robot control and many synonyms may be defined for each action. The second group contains unused words and short phrases. It is the biggest group (vocabulary) with tens of thousands items and we can define names of new actions from this group. The names of learned tasks defined by user are the third group of words in the dialogue scenario. The fourth group contains titles for built-in activities like robot calibration, learning initialization or defining a new name for the most recent operation. Items from this group cannot be used in newly defined commands, though.

The process of learning a new function starts when the operator says the built-in command "beginning of learning". Any known commands issued afterwards are memorized until "end of learning" command is given. A newly defined task has to be given a name. The new name should consist of one previously unused vocabulary word or unused combination of several words (for example "take it + and + move up"). This simple strategy allows to define new robot's tasks just using voice, without keyboard or mouse. It is possible to use any previously defined tasks to compose a new and more complex task.

The picture 3 shows a captured screen of the designed system's GUI. There are buttons representing the basic voice commands on the left side. The majority of the screen is taken up by camera view showing highlighted significant points and detected objects.

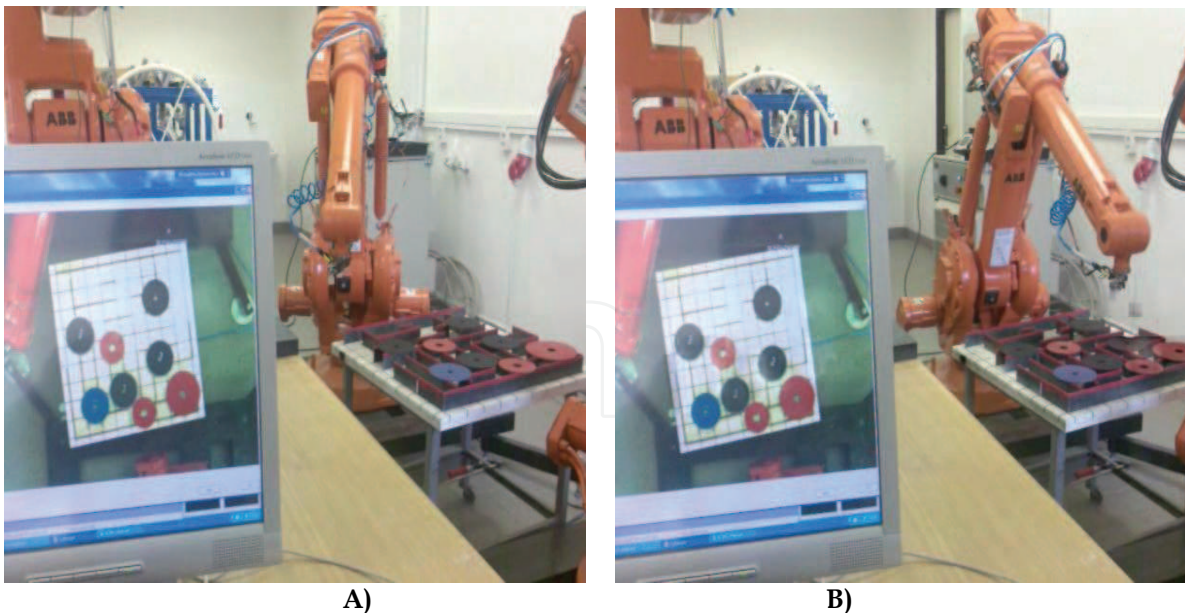


Fig. 4. Scene capture and object detection: A) initial shot with arm outside of view, B) arm carrying out a task

During the dialog some basic logic rules have to be respected. When the objects on the working scene are being analysed, the robot's arm is moved out of scene first (fig. 4) to

avoid object confusion and occlusion. The system stores the last detection result in the scene database.

The whole dialog system is event-driven. We can categorize the events into three fundamental branches: operator events, scene manager events and device events.

3.1 Operator events

Operator events usually occur in response to operator's requests. For example, commands which are supposed to cause robot's movement, object detection, new command definition or detection of a new object. This kind of event can occur at any time, but the dialog manager has to decide if it was a relevant and feasible request or if it was just a random speech recognition error.

Although the acoustic conditions in robotic applications usually involve high background noise (servos, air-pump), the speech recognizer works usually with over 90% recognition score. If the operator says a wrong command or a command out of context (for example, the operator says "drop" but the robot doesn't hold anything) then the event manager asks him or her for a feasible command in the stead of the nonsensical one.

3.2 Scene manager events

This sort of event occurs when the scene manager detects a discrepancy in the scene. For example when the operator says "move up" and the robot's arm moves all the way up until the maximum range is reached. When this happen a scene event is generated and the system indicates that the top position was reached.

Other scene event occurs when the operator wants to take up an object, but the system does not know which one because of multiple objects detected on the scene. This event generates a query to the operator for proper object specification.

3.3 Device events

These events are produced by external sensors and other components and devices connected to the system. They are processed in the event manager where corresponding action is taken. The response manifests itself in the form of a request for the operator, or more often causes a change in robot's behaviour.

The difference between scene manager events and device events is that scene events are generated by the system itself (based on a known scenario, robot geometry, object shape and position). They are computed and predictable. On the other hand, device events' time cannot be exactly predicted before they actually happen.

3.4 Examples of dialog

For a simpler robot orientation and navigation the positions on the scene are virtualized. They are named after the Greek letters like "Position α " (alpha) or "Position β " (beta). These virtual positions may be redefined to suit the operator's needs. A blind-area may also be defined and it is completely omitted from any image processing and anything in this area is completely ignored.

As an example (see figure 3.) the robot can grab all the black disks and move them to some other place on the scene. This place is defined as "Position alpha" and the "blind area" is set up on the same coordinates. After this the operator starts an example dialog:

"*Start recording new command.*" ...this is operator's command (italic text)..
"I'm recording" ...the system says (bold text)..
 "Search black disks"
"I'm searching ... Four disks were found"
 "Move on first"
"I'm moving ... Done"
 "Take it."
"Ok"
 "Move on position alpha."
"I'm moving ... Done"
 "Put it"
"Ok"
 "Stop recording."
"I stop the recording. Please, say new command"
 "Search" "Disks" "Done"
"New command is entered and named: Search disks. Is it right?"
 "Yes"

...now, the newly defined command may be used...

"Repeat command"

"Enter command"

"Search disks"

"OK"

...now the system repeats the command until no more disks are found...

"No object found. Repeating done."

...now all the disks on the scene are transported into position alpha...



Fig. 5. a) The initial scene.

b) The robot grabbing a target disc.

The robot finds the remaining three disks and puts them into the selected area. If no disk is found the robot interrupts the execution of the given command and waits for a new command. This is shown in figures 5 and 6.

The figure 7 shows an operation where the system finds a “small red disk”, grabs it and puts it onto a chosen “black disk”. The navigation of the arm relies only on the image processing results.

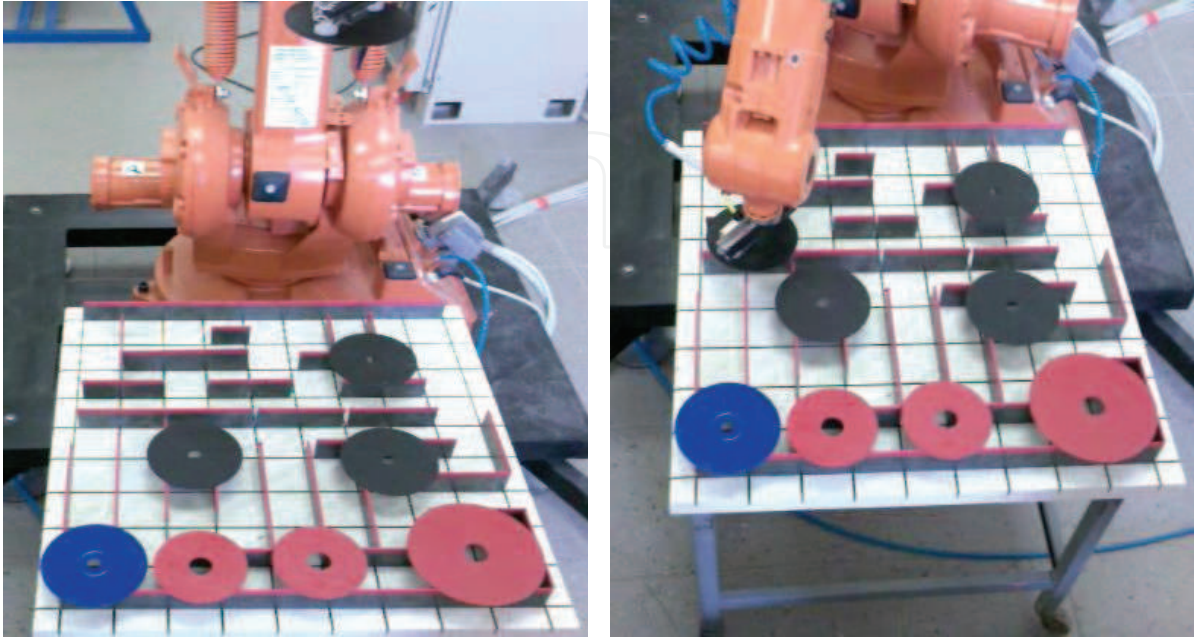


Fig. 6. The robot lifting a disk, moving it around and placing it in a desired position.

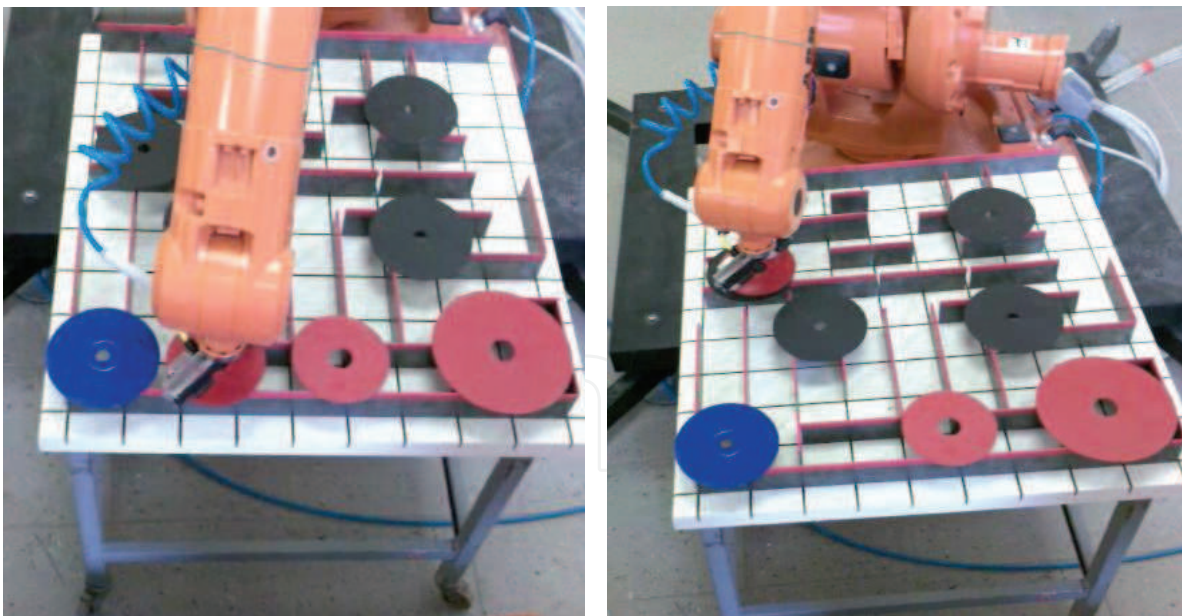


Fig. 7. The robot grabbing another disk and stacking up a pile.

4. Conclusion

The system is especially usable as an accessory robot control interface for assistant and second-rate operations. The designed prototype cooperates with only one specific industry

robot (ABB) so far but the robotic control module may easily be extended to support other robots (Katana, mobile robots, etc.) as well.

The system offers robot control and robot task programming even to people without explicit programming knowledge. It is sufficient for the operator to know the Czech voice interface of the presented system.

The system is able to “memorize” issued commands and reproduce tasks. The designed dialogue strategy was verified using a real robot in real conditions.

The fusion of computer vision, voice recognition and robot control is quite challenging but it looks promising. The development itself is rather complicated as it requires knowledge from many different areas of science.

Employed computer vision greatly simplifies robot navigation as the user actually sees the system’s “understanding” of the scene. This also allows for a much better utilization of voice control.

Contemporary computer hardware seems to be adequate for the demanding operations involved, but the system may still require distributed computing (to achieve reasonable response times and user comfort).

The presented prototype serves as a base for further development. The system is planned to use 3D vision as well as arbitrary object detection and description to become fully 3D-aware and needing as little user aid as possible.

5. Acknowledgement

This work has been supported by the Grant Agency of the Czech Republic (grant no. 102/07/P455) and the internal grant IG FM TUL 2007/002.

6. References

- Nouza J. (2000). A Czech Large Vocabulary Recognition System for Real-Time Applications. In *Text, Speech and Dialogue* (eds. Sojka, Kopecek, Pala) Springer-Verlag, Heidelberg
- Nouza, J., Nouza, T. (2004). A Voice Dictation System for a Million-Word Czech Vocabulary. In: *Proc. of ICCCT 2004, Austin, USA*
- Holada, M. (2004). The experiences and usability of distributed speech recognition system DUNDIS. In: *Proc. of 14th Czech-German Workshop “Speech Processing”, Prague, Czech Republic*, pp. 159-162,
- Hanika, J, Horak, P. (1999). Text to Speech Control Protocol. In: *Proc of the Int. Conf. Eurospeech'99, Budapest, Hungary*
- Sonka, M., Hlaváč, V., Boyle, R. D. (1998). *Image Processing, Analysis and Machine Vision*. PWS, Boston, USA
- Cerva, P., Nouza, J. (2007). Design and Development of Voice Controlled Aids for Motor-Handicapped Persons, In: *Conference of the International Speech Communication Association (Interspeech 2007), ISSN: 1990-9772*



New Developments in Robotics Automation and Control

Edited by Aleksandar Lazinica

ISBN 978-953-7619-20-6

Hard cover, 450 pages

Publisher InTech

Published online 01, October, 2008

Published in print edition October, 2008

This book represents the contributions of the top researchers in the field of robotics, automation and control and will serve as a valuable tool for professionals in these interdisciplinary fields. It consists of 25 chapters that introduce both basic research and advanced developments covering the topics such as kinematics, dynamic analysis, accuracy, optimization design, modelling, simulation and control. Without a doubt, the book covers a great deal of recent research, and as such it works as a valuable source for researchers interested in the involved subjects.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Miroslav Holada and Martin Pelc (2008). The Robot Voice-control System with Interactive Learning, New Developments in Robotics Automation and Control, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-20-6, InTech, Available from:

http://www.intechopen.com/books/new_developments_in_robotics_automation_and_control/the_robot_voice-control_system_with_interactive_learning

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen