Marek ADAMCZYK, Piotr PRZYSTAŁKA, Adam TOMANEK
Department of Fundamentals of Machinery Design,
Silesian University of Technology, Gliwice

THE RAPID CONTROL PROTOTYPING PLATFORM FOR INSPECTION ROBOTS

Key words

Rapid control prototyping, inspection robots, embedded systems, Microsoft® Robotics Studio, Visual Programming Language.

Summary

The scope of this paper focuses on the hardware/software platform that can be applied for rapid control prototyping of such mechatronic systems as mobile robots. Putting together the PC/104 technology, low-cost data acquisition module and Microsoft® Robotics Studio (MRS), the authors have obtained the rapid control prototyping platform that is easy and relatively inexpensive to adapt for developing inspection mobile vehicles. The first objective of this paper is to present a framework for rapid control prototyping of behaviour-based systems in the case of inspection robots. The hardware/software architecture presented here is implemented on the AMIGO robot. This robot is employed for the visual inspection of ventilation ducts. The robot is able to operate in several modes including manual, autonomous, and training.

Introduction

In recent years, autonomous mobile robots, especially ones for inspection must be specialised in monitoring and supervising many areas of human living spaces [10]. Typical examples of such objects are airports, railway stations, refineries, coal-mining shafts, military objects, and many others. In this paper, the authors focus their attention on ventilation duct inspection using a mobile robot to assist in the detection of faults (mainly dust pollution). The visual inspection of such systems is currently performed manually by human operators watching real-time video footage for hours. It is a very boring, tiring, and tedious. Human operators would benefit enormously from the support of an inspection robot able to advise just in the unusual condition. On the other hand, a robot might act autonomously in unknown environments gathering essential data.

Mobile robots that are planed to be used in such situations are usually composed of dedicated hardware and software components [3, 4, 12]. In a large amount of cases, when the design of a robot is still in an early development phase, hardware and software equipment should allow rapid prototyping of a control system. The control architecture presented here may be used in all the stages of control system development. In this project, the authors, using published information, decided to use a behaviour-based control scheme that is known as the "*co-operative/competitive*" or "*Brooksian*" approach [1, 2, 9, 11, 12]. The issues mentioned in this paper are the subjects of more detailed description in other articles of the authors [5, 7, 8].

The paper is organised as follows: In Section 1 the authors present the main factor causing the need to elaborate a platform for rapid control prototyping. In Sections 2, 3 and 4 a much more detailed description of the designed platform is given. In Section 5 there is included an exemplary application of the proposed hardware and software environments. The paper ends with conclusions and future work.

1. Mechanical carrier

The main assumption of the project is that the inspection robot should drive in horizontal and vertical canals of ventilation systems. There are many possible solutions of this problem; however, the authors have selected two main approaches for further research. One of them was a four-legged walking platform with magnetic feet (Fig. 1a). The other was a wheeled platform with magnetic wheels (Fig. 1c). The authors [5] have finally chosen a wheeled platform for further research. 3D CAD systems were used for designing and simulation the kinematics and dynamics of robot prototypes. They have assumed many problems not possible to observe in 3D simulation systems. The authors decided to develop the real model of a wheeled platform (Fig. 1d). They collected a lot of simulating data in real ducts. All of the information gathered with the first model of a wheeled platform was used in order to design the second 3D model of an inspection robot and the final robot prototype (Fig.1e, f).

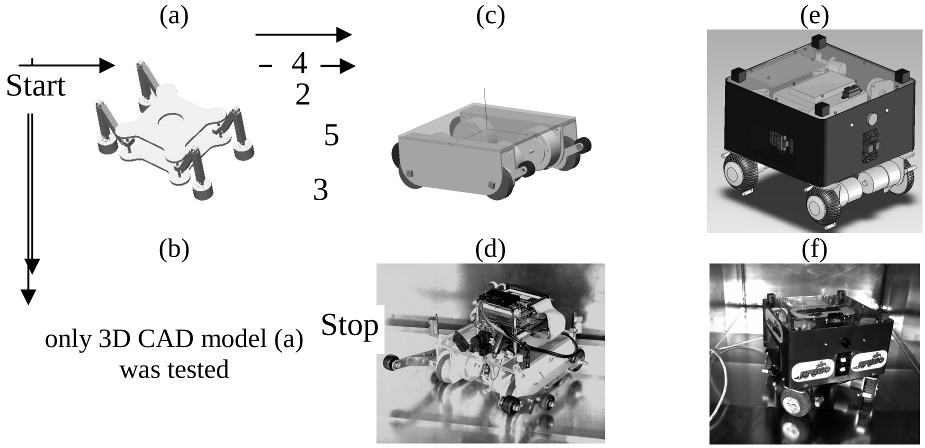

Fig. 1. Designing path of the mechanical carrier of the robot

2. Developing of the hardware layout

As demonstrated in the previous section, the construction of the robot has been modified from a four-legged walking robot (twelve servo-mechanisms) to a wheeled robot (at first with four and next with eight DC motors). This was the main reason that the PC/104 technology was chosen to be used for controlling the motions of the robot. The current version of the low/high-level hardware is described below.

2.1. Low-level controller

The low-level controller allows driving four DC motors mounted in wheels and four step motors used to rotate the second axis of wheels. Every wheel has two degrees of freedom (DOFs) – one for rotation and one for left/right movement. The motor currents were limited to a selectable range from 0.5 to 3.0 Amps per motor in 0,1 Amp increments. A controller can be powered from a supply voltage from +12 to +50 VDC. A variety of sensors was applied to make possible operation in ventilation ducts. Different ranges of infrared distance sensors were selected for this project for measuring the distance to obstacles. The authors also used acceleration sensors for the acceleration of gravity measurements and a gyroscope sensor for rotation measurements. The robot was also equipped with sensors for the internal condition such as electric energy consumption.

2.2. High-level controller

The main module of the high-level hardware layout is a PC/104+ motherboard (Fig. 2) with power supply DC 5V/1.4A. It is equipped with an

AMD™ LX800 500 MHz processor (Ethernet interface, 2 x USB 2.0, RS-232, PCI bus), 512 MB RAM, 4 GB Flash HDD. As one can read in the previous paper of the authors [8], the second component was DMM-32X-AT, a PC/104-format data acquisition board. This configuration was very expensive, and power consumption was unacceptable for real-world industrial applications. Therefore, the authors decided to make use of the low-cost USB-1208FS data acquisition module with a full set of analogue and digital I/O features. It allowed having eight analogue inputs with 12-bit resolution; 50 kS/s maximum sampling rate; two analogue outputs with 12-bit resolution; sixteen lines of digital I/O; one 32-bit counter/timer for general purposes. The robot was also equipped with the IP camera and USB WiFi card.

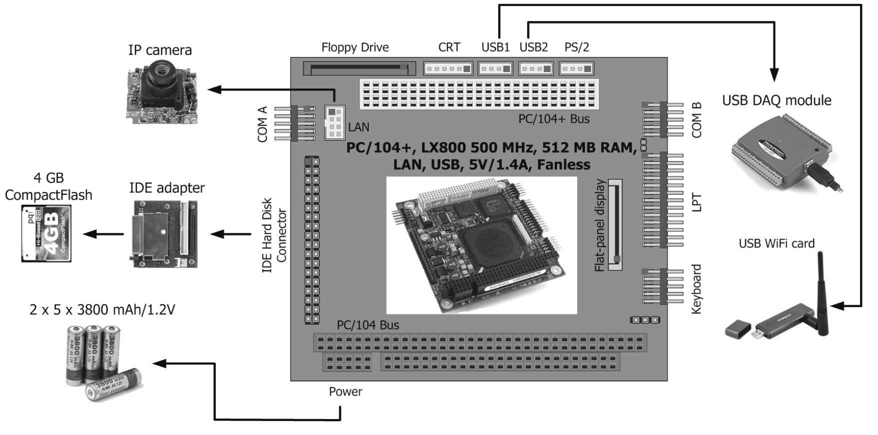

Fig. 2. Main parts of the high-level hardware platform of the AMIGO robot

3. Software components

Figure 3 shows a scheme of the discussed software components implemented on the AMIGO robot and on the PC remote unit. Windows® XP Embedded with Venturcom RTX (hard real-time extension) was employed to guarantee that the response-time determinism requirement is absolute. The authors have considered also such RTOS as RTLinux, RTAI; however, for the reasons that MRS requires Windows-based operating system with .NET environment and services-oriented runtime, the first approach was finally adapted.

With MRS, most current robotic applications are developed using C# language (using Microsoft® Visual Studio). Remotely connected scenarios enable operators to communicate from a PC to the robot through a TCP/IP protocol by a wireless Ethernet link. The developed programs can also be executed natively on the robot running Microsoft® Windows. It enables fully autonomous operations of the robot.

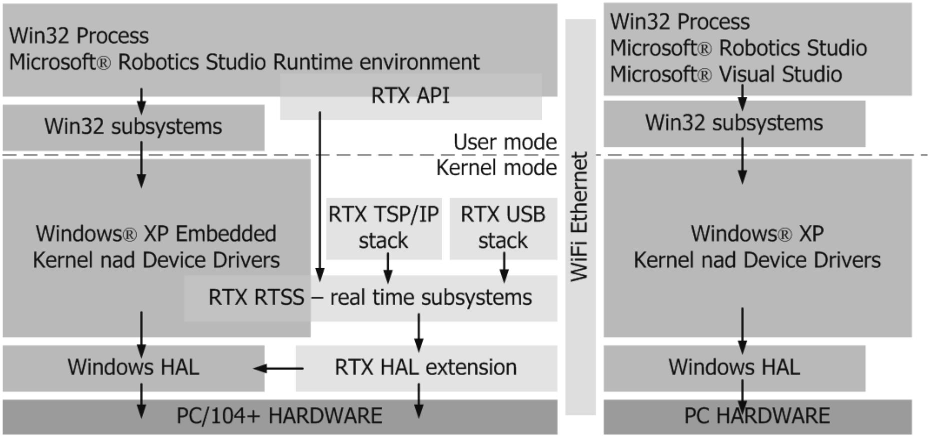

Fig. 3. Main software components installed on the robot and PC remote unit

4. Visual Programming Language

For rapid prototyping or code development purposes, a graphical data-flow-based programming is applied. It is realised using Microsoft® Visual Programming Language (MVPL).

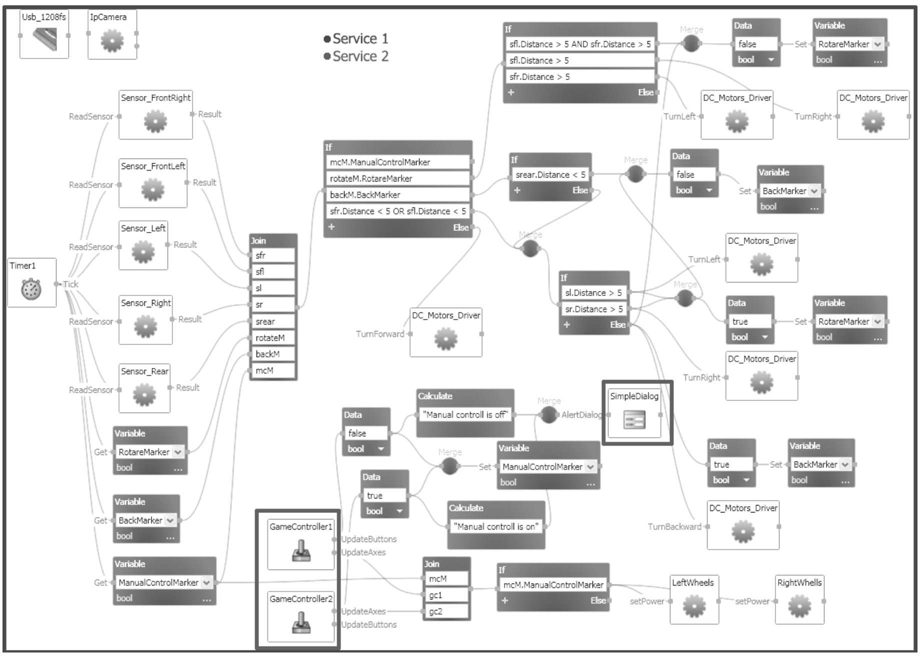

Fig. 4. Sample Microsoft® Visual Programming Language diagram for manual control and for simple wall-following behaviour

MVPL data-flow consists of a connected sequence of activities represented as blocks with inputs and outputs [6, 13]. The main activity block implemented by the authors in C# language is a driver module for the USB I/O board (Usb_1208fs). There are also other blocks included offered by MRS such as: IpCameras, GameControllers, Timers, etc.

Figure 4 illustrates an exemplary MVPL diagram that can be used for manual and autonomous control of the robot. The presented control application is composed of two subservices. The first service is located on the robot, whereas the second one is put on the standard PC-based machine (Fig. 4). Theses programs are communicated via TCP/IP protocol. Manual controllers (GameControllers) are used by an operator in order to select the robot mode (manual or autonomous) or to control its motions. There is also SimpleDialog block for generate warnings and alerts on a display screen watched by an operator. Other blocks (Join, Timer, If, Data, etc.) are used for the reasoning of the robot.

5. Exemplary application

The authors have tested the hardware and software platform by using the AMIGO inspection robot in real ventilation ducts located in their laboratory and in a public building (Fig. 5). The robot uses magnetic force for driving along steel pipes of ventilation ducts. The control system is able to control eight DOFs of the robot.

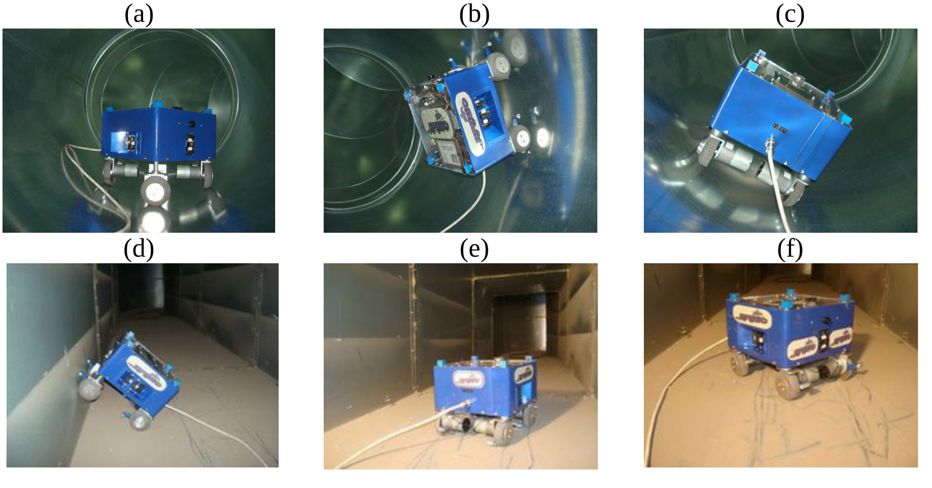

Fig. 5. The AMIGO robot during the inspection of ventilation ducts located in our laboratory (a-c) and in a public building (d-f)

The application generated from a sample MVPL diagram (see Sec. 4) enables the typical capabilities of the AMIGO robot: remote controlling by human operator (using manual controller), monitoring different robot parameters (robot states, real-time video footage) and enabling fully autonomous operations (wall-following behaviour). It is also possible to collect data needed for training the behaviour-based controller.

6. Conclusion and future work

The authors have presented the hardware/software platform that may be used for rapid control prototyping of such mechatronic systems as mobile robots. Putting together the PC/104 technology, low-cost data acquisition module, and Microsoft® Robotics Studio, it is possible to have a hardware/software means that are straightforward and relatively inexpensive to adapt for developing the control systems of inspection mobile robots.

The authors plan to apply a personal digital assistant (PDA) with an USB host port instead of using PC/104-Plus technology. It will allow obtaining hardware with minimal power consumption requirements.

Acknowledgments

Scientific work financed by the Ministry of Science and Higher Education and carried out within the Multi-Year Programme "Development of innovativeness systems of manufacturing and maintenance 2004-2008".

References

1. Arkin R.C.: Neuroscience in motion: the application of schema theory to mobile robotics. In Visuomotor coordination: amphibians, comparisons, models and robots (ed. J.-P. Ewert & M. A. Arbib), 649–671. New York: Plenum, 1989.
2. Brooks R.A.: A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, Vol. RA-2, 1, 1986, 14–23.
3. Holland J.M.: Designing Autonomous Mobile Robots. Elsevier Inc., 2004.
4. Kozłowski K., Krysiak K.: Control systems for mobile robot, Foundations of Control and Management Sciences, 5, 39–58, 2006
5. Moczulski W., Adamczyk M., Przystałka P. and Timofiejczuk A.: Mobile robot for inspecting ventilation ducts. Recent Advances in Mechatronics, Springer Berlin Heidelberg, 47–52, September 2007.
6. Morgan S.: Programming Microsoft® Robotics Studio. Microsoft Press, Washington, 2008.
7. Panfil W., Przystałka P. and Adamczyk M.: Behavior-based control system of a mobile robot for the visual inspection of ventilation ducts. Recent Advances in Mechatronics, 62–67, Springer Berlin Heidelberg, September 2007.

8. Przystałka P. and Adamczyk M.: EmAmigo framework for developing behavior-based control systems of inspection robots. Recent Advances in Mechatronics, Springer Berlin Heidelberg, 37–42, September 2007.
9. Scheutz M. and Andronache V.: Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems. Systems, Man, and Cybernetics, Part B, 2377–2395, 2004.
10. Shuzhi Sam Ge, Lewis F. (Eds.): Autonomous Mobile Robots. Sensing, Control, Decision, Making and Applications. CRC Press Taylor and Francis Group, 2006.
11. Zielińska T., Chmielniak A.: Sterowanie zachowaniami zespołu robotów mobilnych, VI Krajowa Konferencja Robotyki, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1998, 307–314 (in Polish).
12. Zieliński C.: Formal approach to the design of robot programming frameworks: the behavioral control case. Bulletin of the Polish Academy of Sciences – Technical Sciences. 53, 1, 2005, 57–67.
13. Microsoft Robotics Studio: An Introductionhttp://msdn.microsoft.com/pl-pl/library/bb483024(en-us).aspx, 10.10.2008.

Reviewer:
Cezary ZIELIŃSKI

Platforma do szybkiego prototypowania układów sterowania robotów inspekcyjnych

Słowa kluczowe

Szybkie prototypowanie układów sterowania, roboty inspekcyjne, systemy wbudowane, Microsoft® Robotics Studio, język Visual Programming Language.

Streszczenie

W artykule zaprezentowano sprzętową i programową platformę, która może byś stosowana do szybkiego prototypowania takich układów mechatronicznych, jak roboty mobilne. Połączenie technologii PC/104 oraz tanich modułów akwizycji danych, jak również środowiska Microsoft® Robotics Studio umożliwiło uzyskanie środka ułatwiającego szybką adaptację i rozbudowę robotów. Głównym celem artykułu jest prezentacja narzędzi do szybkiego prototypowania systemów sterowania robotów mobilnych opartych na zachowaniach. Prezentowana architektura jest zaimplementowana na robocie AMIGO. Robot ten służy do wizualnej inspekcji kanałów wentylacyjnych. Robot działa w następujących trybach: ręcznym, autonomicznym i treningowym.