PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 41, NO. 3, PP. 239–249 (1997)

A LOW-COST ROBOT CONTROLLER AND ITS SOFTWARE PROBLEMS

Gábor TEVESZ, István BÉZI and István OLÁH[1]

Department of Automation
Technical University of Budapest
H–1521 Budapest, Hungary
Phone: 36 1 463-2870, Fax: 36 1 463-2871
E-mail: tevesz@aut.bme.hu, bezi@aut.bme.hu, olah_i@aut.bme.hu

Received: July 4, 1997

Abstract

In recent years the need for advanced robot control algorithms for industrial robots has grown. The development of a low-cost robot controller to support the development, implementation and testing of those algorithms which require high computational power was targeted. This paper deals with the requirements of an experimental controller that can be connected to a NOKIA-PUMA 560 robot arm. It explains the IBM PC compatible host and the TEXAS Digital Signal Processor (DSP) based hardware. On the host computer the UNIX-like QNX real-time operating system is used. In the current phase of development the robot controller works with the classical decentralised joint control based strategy. The Advanced Robot Programming System (ARPS) explicit robot programming language is implemented.

Keywords: robot control, multiprocessor systems, IBM PC, DSP, QNX, ARPS.

1. Introduction

Numerous advanced robot control algorithms, such as computed torque technics (nonlinear decoupling and decentralised PID controllers), resolved motion acceleration control, hybrid position and force control (operational space formulation), model reference adaptive control and fuzzy control, are currently undergoing research and experiment. Such algorithms are increasingly in demand in the workplace. The hardware and software of industrial robots are closed systems, with an unsatisfactory computational power and thus the implementation of advanced control strategies is not feasible.

---

[1] The research work was supported by the Hungarian research Fund (OTKA) under the terms of grant No. T 016855

2. The Requirements of the Robot Controller

Advanced robot control algorithms are based on the nonlinear dynamic model of the robot arm with a driving torque of:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + (\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q}) = \tau, \qquad (1)$$

where $\mathbf{H}$ is the inertia matrix, $\mathbf{C}$ is the centripetal, Coriolis and friction effects, $\mathbf{D}$ is the gravitational part and τ is the driving torque. An over-riding problem in the implementation of such algorithms is that the driving torque needs to be computed within 1–10 ms. This computation requires multiprocessor architecture and nonrecursive algorithms are favoured where parallel computing is possible (LANTOS, 1991 and SOMLÓ et al., 1997).

The low-cost experimental robot controller was initially developed for a six degrees of freedom NOKIA-PUMA 560 robot arm. The requirements of the controller hardware were as follows:

- modular architecture,
- extendibility,
- easy system development possibility,
- interfacing several sensor processors possibility,
- changing the host computer possibility.

The robot controller should:

- be an open system,
- have a modular (layered) structure that does not reduce efficiency,
- be based on such programming languages and methods that they guarantee software portability.

3. The Hardware

Examining the possibilities of the development of such systems the IBM PC based multiprocessor architecture was chosen (BÉZI and TEVESZ, 1994). Two robot controllers were built for the cooperating departments. One works at the Department of Automation while the other is used at the Department of Process Control.

Regarding the above requirements, the aim was to develop an open system where the implementation and testing of different control algorithms is easy. Capitalising on the benefits of commercial products, the available parts of the control system were chosen for their speed, serviceability and interchangeability. The whole control system was built on the IBM PC basis because of the many standardised, easily obtainable modules and because of the wide-spread usage of these machines in the field of process control. The multiprocessor core of the system and the interface to the controlled process was target specific design. The result of this development is shown in *Fig. 1*.

The present host computer is based on an i486DX2/66 main board. The advantage of this choice is the extensibility of the performance by simply upgrading the host processor without major changes in the software system being developed. The link between the modules of the system is the ISA bus. This is the only bus system in the 12 year history of IBM PC that has survived all other improvements in more modern announced bus systems (EISA, VESA, VLB, PCI). Because of the number of used modules the host computer consists of two racks connected with ISA bus expansion cards without reducing the speed of the communication.

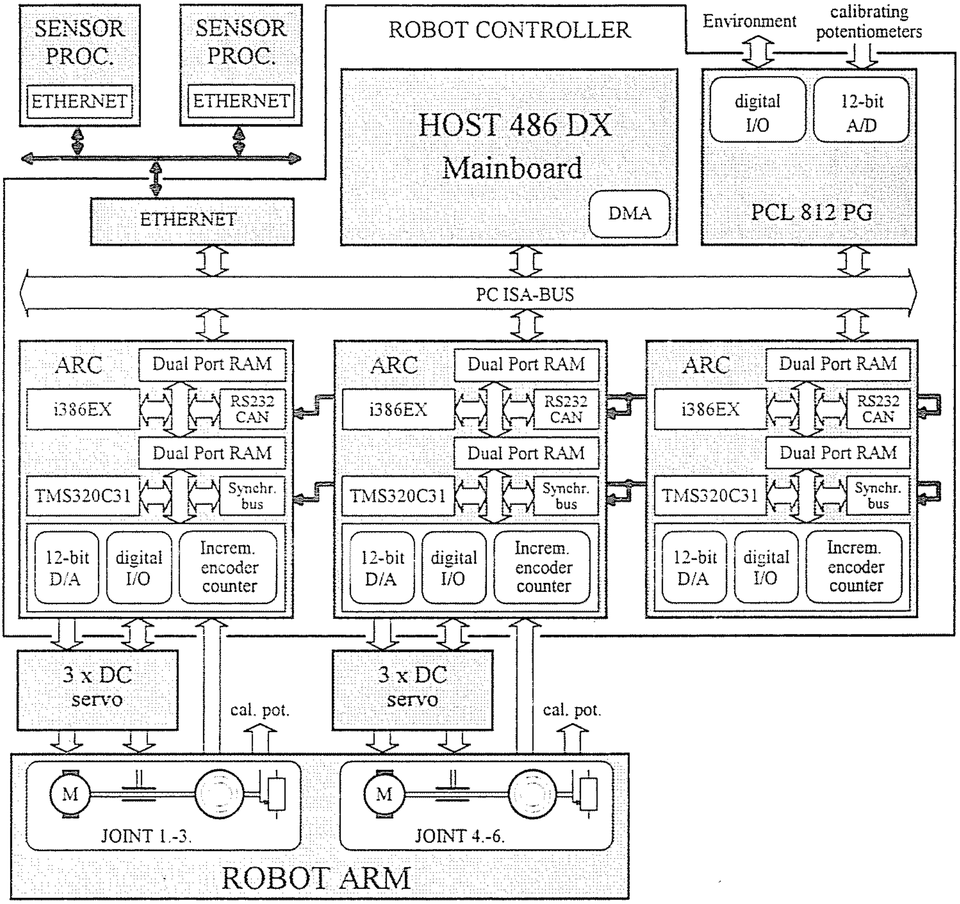

Fig. 1. Configuration of the robot controller

The connection between the host and the other intelligent units is provided by a Local Area Network. At present this is built up with a 10 Mbit/sec Ethernet line but there are no difficulties in changing it to the compatible type of 100 Mbit/sec speed on demand. The benefits of this choice are: standardised modules, developed continuously, newer and newer products are compatible with previous ones and it provides the spatial distribution of the units.

The connection between the host and the passive environment is provided by a high speed, high performance, multi-function data acquisition card (PCL-812PG). The calibrating potentiometers of the robot arm are connected to the programmable gain analog inputs. In addition the data acquisition card supports connecting of 16 digital input and 16 digital output lines that provides more control tasks to be made.

The algorithmic part of the robot control and the direct control of the arm is fulfilled by the self developed Advanced Robot Controller (ARC) card. This is the most important part of the system so the features are explained in detail. This card is a multiprocessor card which communicates with the host over the ISA bus, with each other over the CAN bus, a high speed serial bus and over a special parallel (synchronising) bus. This card provides all the signals for the external DC servo amplifiers and receives the sensor signals of incremental encoders in the robot arm.

To increase the computational performance a high-speed transputer card can be implemented in the system. The task of this card depends on the particular control algorithm. The aim is to communicate with this card over the ISA bus providing interchangeability with another, more suitable, task specific card regarding the unified communicational system. In the present system a third ARC card is used for the computational needs of the advanced control algorithm. The first two ARC cards are used for the direct control of axes.

The six joints of the robot arm are directed by the self developed DC servo amplifiers. These amplifiers work over 18 kHz switching frequency with pulse width modulation. The set point signals for current come from the ARC cards in analog form and the analog PI controllers of the servo amplifiers perform the real values (and limits on demand).

The most important part of the experimental robot control system is the ARC card. The block scheme of this module is shown in *Fig. 2*. Every card consists of two microprocessors: the so called preprocessing unit is an i386EX microcontroller while the second, so called joint processor, is a TMS320C31 DSP. On each card 4 whole featured interface is available for electronic drives (joints or axes). Considering the complex task of robot control and depending on the implemented algorithms the use of two or

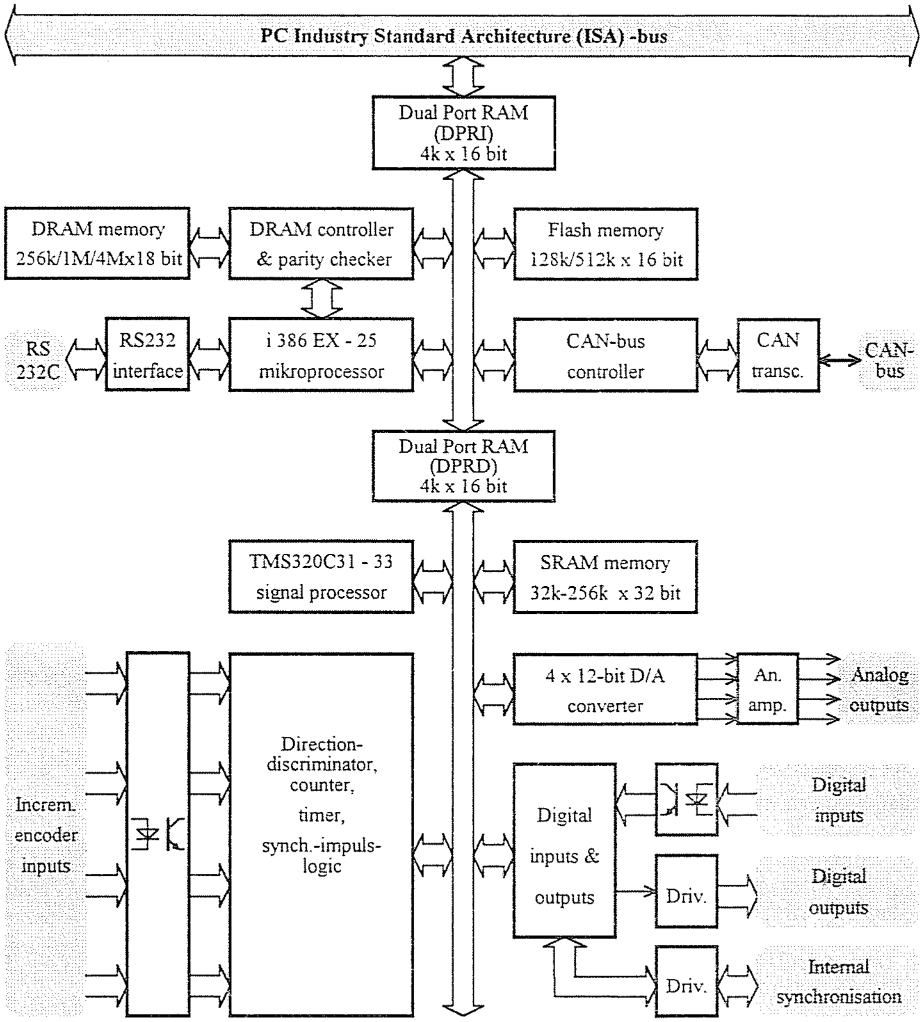

Fig. 2. Block scheme of the ARC board

three ARC cards is planned. This way each card provides the connection and control task for three or two joints.

The main blocks of the cards are as follows:

The preprocessing unit is the new embedded microprocessor by Intel (i386EX). This device provides a 100% compatible environment with PCs integrating the basic peripheral functions of a main board. This 32 bit, 80386 based microprocessor serves not only with the MMU for the possibility of Virtual Protected Mode but contains special enhancements like the missed watch-dog circuit. The memory of this unit consists of a flash block (256 kbyte or 1 Mbyte) and the usual DRAMs (the capacity depends on the used SIM modules: from 512 kbyte up to 8 Mbyte). The non-volatile chip contains at present a BIOS but the implementation of a whole ROM-DOS system is planned which would make the development and testing of different robot control algorithms much easier.

The joint processing units are high-speed CMOS 32-bit floating-point single-chip Digital Signal Processors (DSP) - TMS320C31. These DSPs have the highest performance in the system with a capability of 16 MIPS and 32 MFLOPS. The tasks of these joint processors are:

- taking the position signals and calculating the speed and acceleration,
- credibility check using the null impulses,
- supervising the position, speed and acceleration limits,
- producing the current set point values for servo amplifiers,
- providing the synchronization in starting and stopping the axes.

The high performance of the joint processor is supported by the high-speed static RAM (32–128 kword). The required access time (18–20 nsec) is unreachable in the case of non-volatile memory so this device uses the built in Boot Loader to load its own program system from the flash of the preprocessing unit over the (dual port) DPRAM. The flash memory of the preprocessor contains the programs for both processors on the card. In order to the highest possible response time the signals of the position, speed and acceleration sensor incremental encoder are connected directly to the DSPs through the intelligent interfaces (direction discriminator, counter and null impulse logic) realised by tacho processors of type TC3005H. The analog circuits for current set point and the servo amplifier controlling digital input and output signals are also connected to the joint processor directly. A three-wire synchronisation channel provides for simultaneous movement of the axes connected to different ARC cards.

Communication Channels inside the System

The most important questions in time of the design were the choice of proper communicational channels. The low throughput of these connections would limit the performance of the responsible units.

These channels are shown in *Fig. 3*. The broken connections are for the most important development possibilities that are not implemented in the present system.

The preprocessing units are functionally set between two other processors (the host and the joint processor) and the communication in both directions has a main role in the functionality. Since both of the connections have a high capacity demand the highest throughput has been chosen here: the links are high-speed dual port RAMs (DPRI and DPRD blocks in *Fig. 2*). This realisation provides the maximal speed communication because the communication is delayed only in the case when both sides access the same cell simultaneously. The throughput of the channels between the host and preprocessor using the built in 16 bit-word access is approximately 5 Mbyte/sec and between the preprocessor and joint processor is 10 Mbyte/sec. In order to attain optimal performance the available interrupt possibilities should be used on every side. 8 built in hardware semaphores per page are available for the data block consistency.

The preprocessors in communication with each other should not use the host (over the other way loaded ISA bus) but these modules are connected via a real multimaster serial Controller Area Network (CAN) bus. This bus is accessible on the back plane of the ARC cards so other external elements of the control system can be connected via CAN bus on demand. The physical speed of 1 Mbit, concerning the message identification and check philosophy of CAN protocol, provides approximately 50 kbyte/sec data throughput. Of course this channel is not for moving huge data blocks but rather for changing intermediate results of robot control algorithms. The multimaster property of CAN bus providing the communication of equal members is essential (however, the traffic can be mastered by one).

The RS232C compatible serial line of the preprocessor has been built only for test purposes in the early development phase.

A special synchronisation bus has been developed for the joint processors providing the simultaneous movement of joints. This bus consist of only three lines with the availability of wired OR (in negative: AND) logic between the processors. Its speed is ca. 8 Mbit/sec.

5. Software

The system was designed to be low-cost so the heterogeneous architecture caused some informatical problems. The requirements were:

- software development and test on host,
- connection with intelligent sensors via network,
- user friendly operator interface,
- software development and test for the preprocessor (i386EX),
- program transfer and running on the preprocessor,
- software development and test for the joint processor (TMS320C31),
- program transfer and running on the joint processor.

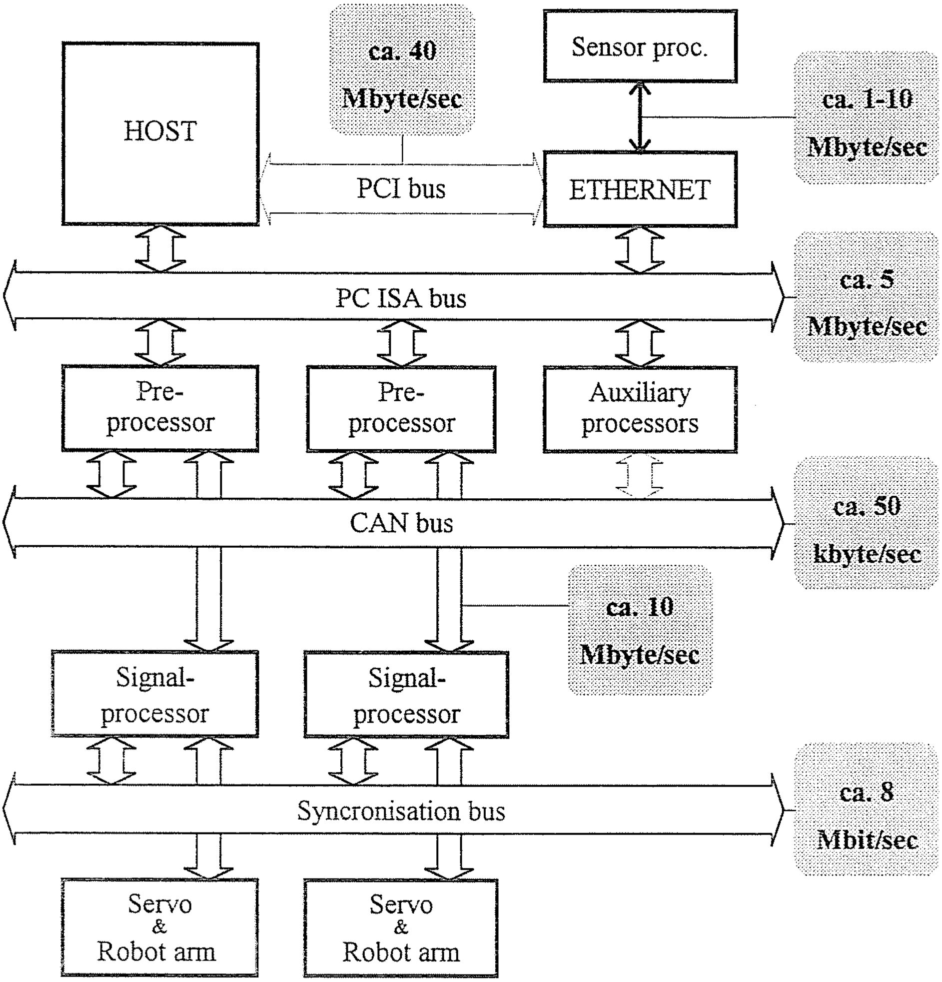

Fig. 3. Communication channels inside the system

In the host computer the QNX real-time UNIX-like operating system was chosen (DODGE, 1992). This provides multitasking, priority-driven preemptive scheduling with three scheduling algorithms following the IEEE POSIX 1003.4 specification. The QNX is a flexible, distributed (networked) operating system, characterised by microkernel architecture and message-based interprocess communication. This interprocess communication is performed in three ways: as messages (synchronous communication), proxies (special form of messages) and signals (asynchronous communication).

Another communication possibility is shared memory when proxies and signals can provide the synchronisation. The QNX has advanced timing facilities and the interrupts can be handled on a process level. The QNX Windows system provides the user friendly interface.

In the interests of portability and transferability of particular tasks between processors the software development uses only C language (Watcom/Borland/ Microsoft/Texas Instruments). The benefits of this 'broad-spectrum' (high and low level capabilities) language are well-known and C is available for all used microprocessors.

The first phase of software development for ARC card provides a simple BIOS. With the help of this system the program transfer and running is possible through few system calls (software interrupts on i386EX and assembly function calls on TMS DSP). In addition to the previous functions the possibility of character input and output is provided for both processors. Based on the BIOS a developing environment was made which is used for program and test each processor using C language. The preprocessor is able to run programs in DOS EXE format but does not provide the whole IBM PC environment. The startup code of Borland C was changed and the same type of modification was made in case of the Watcom C compiler. In order to achieve proper functionality of the watch-dog circuit the boot routine (c_int00) of the Texas Instruments C compiler for TMS320C3x/C4x processors had to be modified as well. This compiler is available only for DOS so the QNX Rundos (a DOS emulator) was used first. Because of the incompatibility between Rundos and TMS C compiler this method was cancelled (VÖRÖS, 1995).

For the build up and first time check of the whole hardware system the following programs were written for QNX Windows:

- set and measurement of the analog and digital ports on the PCL-812PG card. This program is able for displaying the signals of the potentiometers in the robot arms,
- program transfer and running in the joint processor using the preprocessor only for data forwarding,
- determination of joint position and speed giving out the current set point and sending back the information (including ARC status) to the host.

Based on the experience of test programs low level control software was made which realizes a double loop digital PID controller for speed and position (the third loop is the analog PI controller in the DC servo amplifiers). The communication protocol of the undocumented teach pendant was decoded and the driver for this device was implemented in the software. Other software modules are available for determining the characteristic of potentiometers and calibrating the robot arm (DONÁT, 1996 and KORCSMÁR, 1996.).

The Advanced Robot Programming System (ARPS, the original programming system of the NOKIA-PUMA 560) was implemented in the Department of Process Control (JÁSZ, 1996) and integrated into the software system (KORCSMÁR, 1996).

6. Conclusions

The experimental robot control system in its present form is serving well in the education of robot programming. Considering the present experience for the realization of advanced robot control algorithms the next steps are:

- developing the BIOS system of the ARC card burning some new functions into flash memory,
- speeding up the communication over dual port RAMs,
- working out the communication on CAN bus,
- implementing an operating system subset (utilizing the embeddable capabilities of QNX microkernel architecture) on the preprocessor.

The present experimental robot control system on the Department of Process Control will be used as part of an intelligent control system of a robot with a dextrous hand. The development of the software based on the above architecture is being continued at the Department of Process Control in some new directions like a new communication principle between host PC and ARC control boards, new basic software components, advanced control algorithms using self tuning adaptive control, neural control and fuzzy experts (KLATSMÁNYI, 1996). The architecture and the software of two further subsystems were developed at the Department of Process Control (LANTOS et al., 1997). The first of these new subsystems is the control system of the tendon-driven three-fingered (9 degree-of-freedom) TUB dextrous hand (LUDVIG, 1996a), allowing high-level grasp-planning and implementation (LUDVIG, 1996b). The second new subsystem is a low-cost stereo vision system. The task of the stereo vision system is to collect image information about the robot, the hand and the environment, process it and send the results to the control systems of the robot and the hand. The image processing is based on the theory of projective geometry, statistical object recognition and parameter identification (LANTOS et al., 1997).

References

1.BÉZI, I. – TEVESZ, G. (1994): PC Based Robot Controller for Advanced Control Algorithms. *Proc. of the Joint Hungarian - British International Mechatronics Conference*, Budapest, September 21-23, 1994, pp. 743-748.

2.BÉZI, I. – TEVESZ, G. (1996): Komplett aus Standard-Komponenten. *Elektronik*, 1/1996, pp. 44-48.

3.DODGE, D. (1992): The Continuing Evolution of QNX. *QNX News*, No. 6, pp. 7-15.

4.DONÁT, T. (1996): Kísérleti robotvezérlő élesztése és tesztelése (Setting up and Testing of an Experimental Robot Controller). M. Sc. Thesis at the Department of Automation, Technical University of Budapest (in Hungarian).

5. JÁSZ, G. (1996): Robotprogramozási nyelv kifejlesztése QNX valósidejű operációs rendszerhez (Development of a robot programming language under QNX real-time operating system). M.Sc. Thesis at the Department of Process Control, Technical University of Budapest (in Hungarian).

6. KLATSMÁNYI, P. (1996): Design and Implementation of a New Communication Concept and its Application in the New Version of the Control System of a PUMA Robot. OTKA T 16855/3 Research Report at the Department of Process Control, Technical University of Budapest.

7.KORCSMÁR, A. (1996): Teszt és működtető programok kifejlesztése célhardverekhez (Development of Testing and Operating Programs for Special Hardware). M. Sc. Thesis at the Department of Automation, Technical University of Budapest (in Hungarian).

8. LANTOS, B. (1991): Robotok irányítása (Robot control). Akadémiai Kiadó, Budapest (in Hungarian, 2nd edition 1997).

9. LANTOS, B. – KLATSMÁNYI, P. – LUDVIG, L. – TÉL, F. (1997): Intelligent Control System of a Robot with Dextrous Hand. *Proceedings of the IEEE International Conference on Intelligent Engineering Systems INES'97*, Budapest (being appeared).

10. LUDVIG, L. (1996a): Intelligent Control of Dextrous Robot Hands. *Proceedings of the 5th International Workshop on Robotics in Alpe - Adria - Danube Region RAAD'96*, Budapest, pp. 345–350.

11. LUDVIG, L. (1996b): A Practical Approach to the Choice of Optimum Grasp for Three-fingered Dextrous Hands. *Proceedings of the 27th International Symposium on Industrial Robots ISIR'96*, Milan, pp.459–464.

12. SOMLÓ, J. – LANTOS, B. – CAT, P. T. (1997): Advanced Robot Control. Hungarian Academic Press, Budapest (in press).

13. VÖRÖS, I. (1995): Programfejlesztői környezet kialakítása kísérleti robot vezérlőhöz (Design of Program Developing Environment for an Experimental Robot Controller). M. Sc. Thesis at the Department of Automation, Technical University of Budapest (in Hungarian).