Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms

Skiba Grzegorz, Żabiński Tomasz, Bożek Andrzej
Rzeszów University of Technology
Email: skiba.g@gmail.com, tomz@prz-rzeszow.pl, bozekand@op.pl

Abstract—**This document describes three didactic systems with a Rapid Control Prototyping (RCP) suite based on Scilab/ Scicos and a PC computer. Servomechanisms with wire signal transmission and PID as well as a fuzzy controller are presented. The servomechanism with wireless signal transmission is also described. As the RCP suite based on Scilab/Scicos/RTAI was successfully used in the servo controller development on a PC platform, the concept of a tool-chain for RCP on an embedded platform with ARM processor is discussed. The TS-7300 embedded system with RTAI real-time operating system is described. The changes in Scilab/Scicos needed to interface the generated controller code to TS-7300 are presented. The didactic use and a possible commercial use of the tool-chain is indicated. The didactic use is focused on controllers tuning, friction and long delays influence on servomechanism control and embedded control systems development.**

I. Introduction

Rapid Control Prototyping (RCP) gives a possibility for quick and convenient control strategy verification and iterative controller development. RCP involves a controller simulated in real-time, coupled with a real plant via hardware input/output devices. Nowadays RCP has become a crucial method in developing and testing control strategies within time acceptable by the market. It is also a very popular technology in scientific and didactic laboratories.

RCP requires two components: a Computer-Aided Control System Design (CACSD) software and a hardware with a hard real-time operating system. CACSD includes a broad range of computational tools and environments for: control systems design, real-time simulation, data acquisition and visualization. CACSD should support all the phases of the control system development namely specification, modeling, identification, controller design, simulation, implementation and verification [1], [2]. The most significant element of CACSD in supporting implementation is an integrated code generator which allows a direct creation of a controller code from e.g. a graphical schema. One of the most popular RCP systems is based on the Matlab/Simulink/RTW (Realtime-Workshop) suite [3] which enables creation and compilation of a controller code for different targets. The Matlab/Simulink/RTW commercial product is very popular at universities and in industries. The main advantage of the suite is its convenience and accurate graphical interface. The main drawback is given by its cost. An interesting alternative to Matlab/Simulink/RTW is the free and open-source Scilab/Scicos/RTAICodeGen suite [4]. The alternative is based on Scilab/Scicos, Linux RTAI and RTAI-Lab. The RCP open source environments are especially desirable for didactic purposes because of their low cost. The RCP tool-chain based on RTAI-Lab has been successfully used in Department of Computer and Control Engineering at the Rzeszów University of Technology. The tool-chain is used in Mechatronics, Control Theory, Embedded Systems and Real-Time Control courses. Additionally the tool-chain is also used by students and researchers who engage in the creation of control systems.

This paper describes the existing laboratory equipment for servo control with wire or wireless signal transmission. The examples of using PID and Fuzzy Model Reference Learning Controller (FMRLC) for a servo control are also briefly described in the paper.

The paper is organized as follows: in Section II the Scilab/ Scicos/RTAICodeGen suite is described; in Section III three different didactic control systems for servomechanisms are presented; in Section IV an idea of the RCP tool-chain for embedded system as well as the architecture of TS-7300 em bedded platform with ARM9 processor and FPGA chip is briefly described; and finally in Section V the main results are briefly discussed.

II. Scilab/Scicos/ RTAICodeGen suite

Scilab is a scientific software for numerical computations. It has been developed since 1990 by scientists from INRIA (Institut National de Recherche on Informatique et on Au tomatique) [5] and ENPC (École Nationale des ponts et chaussée) [6]. Scilab provides a rich set of functions for sci entific applications and engineering [7].

Scicos is a Scilab toolbox which allows a graphical design of a control system. The design and simulation of a control schema can be realized directly from the Scicos graphical interface in a comparable manner to Simulink.

Linux RTAI is a hard real-time extension of the Linux Operating System. RTAI has been developed since 1999 by the researches from the DIAPM (Dipartimento di Ingegneria Aerospaziale del Politecnico di Milano) [8].

The RTAI environment has been extended to include a code generator called RTAICodeGen in order to obtain com patibility with RTAI for a hard real-time code from Scicos schemas.

RTAI-Lab is a tool included in an RTAI distribution which provides a framework for designing, building, running and monitoring RTAI-based controllers and real-time simulators. The controllers can be coded manually in languages akin to C/C++ or generated automatically by Scilab/ Scicos/ RTAICodeGen or by Matlab/Simulink/RTW. The RTAI-Lab framework contains block library that allows the interaction of Scicos schem a s with a data acquisition hardware. Drivers for acquisition hardware are provided by the COMEDI project [9] or can be implemented by the user [4]. *Xrtailab* the GUI application included in RTAI-Lab is used for the purpose of data acquisition and monitoring and so as to change controller parameters on the fly.

The generated controller typically runs as a user space hard real-time application on a standard PC computer [1], [10]. Nowadays it is a typical way of using the RCP suit based on Scilab/Scicos and RTAI. As RTAI supports the following hardware architectures:

- x86 and x86_64
- PowerPC
- ARM (some subarchitectures)

it would be convenient to use the RCP suite for Rapid Control Prototyping on a destination embedded hardware [11].

Section IV briefly describes the way to obtain the RCP tool-chain for the TS-7300 embedded system with the ARM920T processor.

III. Control of a DC servo motor in Scilab/Scicos/RTAI-Lab

Three didactic servo control systems developed in Department of Computer and Control Engineering are described in this section. The systems are used for didactic purposes in courses of Mechatronics and Control Theory.

A. Plant description

The plant includes a DC servo motor from Pittman, an HP incremental encoder, an MSA-12-80 power amplifier in current mode from Galil and a KR33 linear stage with a ball screw from THK (Fig. 1).

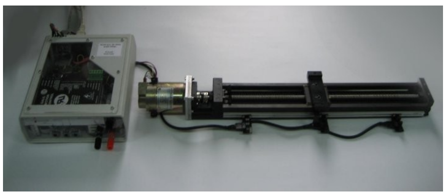

Fig. 1. Laboratory servomechanism

For RCP purposes a general PC computer equipped with Linux RTAI as real-time operating system is used. For wire signal transmission the input/output board RT-DAC4-PCI [12] from Inteco is used. The driver for RT-DAC4-PCI card was developed using COMEDI specification [4]. In wireless communication two transmission modules are used: one connected to the power module and the second connected to the PC computer via the RS232 interface. The transmission modules were constructed using ADuC7128 controllers and TLX2401 radio modules [10].

B. DC servo control with wire signal transmission and PID controller

In order to use the RT-DAC4-PCI card in real-time experiments the driver and communication blocks for Scicos were developed [4]. The blocks for: A/D and D/A conversion, PWM outputs and encoder counters were developed (Fig. 2).

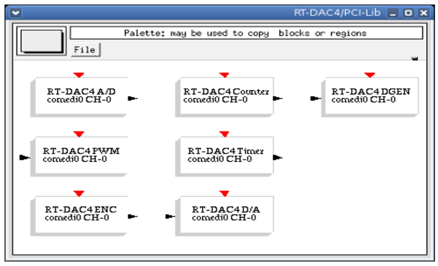

Fig. 2. Communication blocks for RT-DAC4-PCI and Scicos

Control system structure with PID digital controller is presented on Fig. 3.

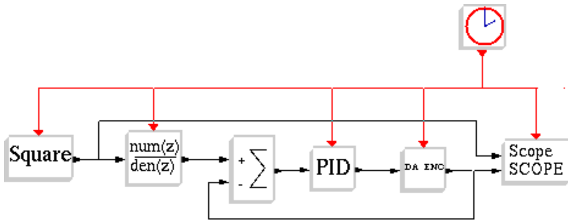

Fig. 3. Control system structure for DC servo control with PID discrete controller

The block DA-ENC includes the D/A conversion block and the encoder counter interface block which provides communication with the real plant (Fig. 4).

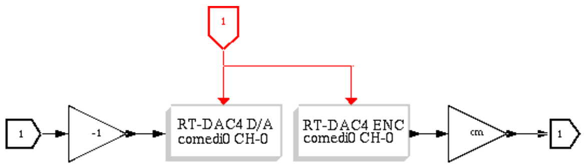

Fig. 4. DA-ENC block internal structure

The PID block internal structure is presented on Fig. 5.

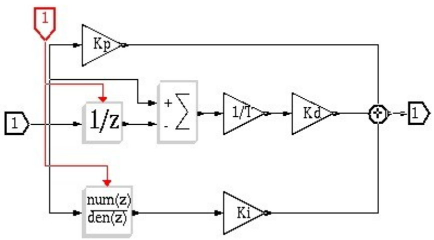

Fig. 5. Discrete PID controller structure

The system is mainly used for didactic purposes. The students perform identification procedure of the real-plant model and perform PD and PID controllers tuning. Many different experiments are performed in order to test the step response parameters and the influence of friction on servo-mechanism systems e.g. steady-state errors, hunting and stick-slip.

A plant model given by a double integrator transfer function is identified via a step response real-time experiment conducted using Scicos schema. The Scilab *datafit* function is used to fit the model to the experimental data. The PD and PID controller is tuned using root-locus design method. In this case, two design data, i.e. gain of the drive treated as double integrator and required settling time, are only needed [13]. The results of servomechanism real-time step response experiment are presented on Fig. 6.

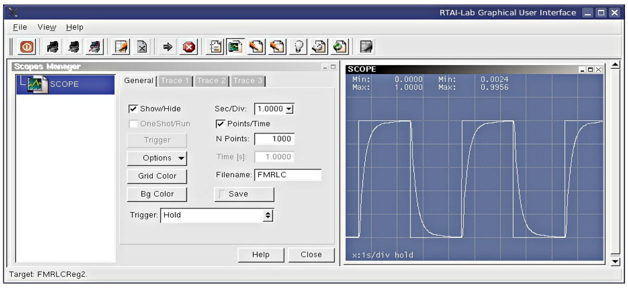

Fig. 6. Servomechanism real-time step response experiment with PID controller - *xrtailab*

The obtained results are similar to the ones which use the RCP suite based on Matlab/Simulink/RTW and RT-CON [12].

C. DC servo control with wire signal transmission and FMRLC controller

A more sophisticated controller for a servomechanism with Fuzzy Model Reference Learning Controller (FMRLC) [14] is also used by the students [4]. The control system consists of the reference model, fuzzy Takagi-Sugeno controller under learning process, fuzzy inverse model, learning mechanism and knowledge-base (Fig. 7). The two inputs to the fuzzy PD controller are the error e and change in error de (Fig. 7). On the basis of the internal knowledge the controller produces control signal u which is fed to the plant (power amplifier). The fuzzy controller is continuously taught during the control process by a fuzzy inverse model, learning mechanism and knowledge base. The learning goal is to keep a plant output (y) as close as possible to the reference model response (y_m). Only the consequents of the controller rules are changed, fuzzy sets remain unchanged.

The Scicos control schema with FMRLC controller is shown on Fig. 8.

The FMRLC block was developed in C language as a computational function for Scicos [4], [7].

The system is used for didactic and research purposes. The students perform experiments with different configurations of FMRLC e.g. simple fuzzy controller in control loop and complete FMRLC structure.

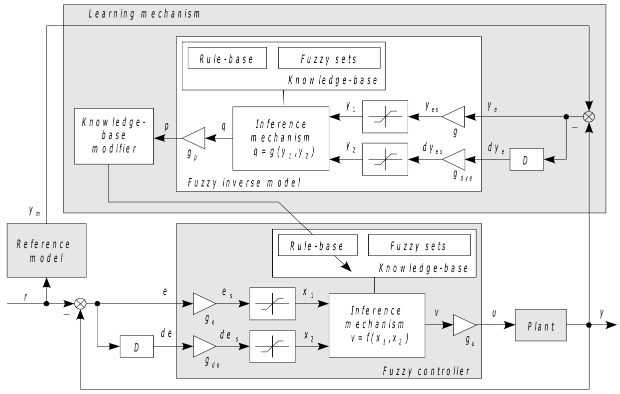

Fig. 7. FMRLC structure [14]

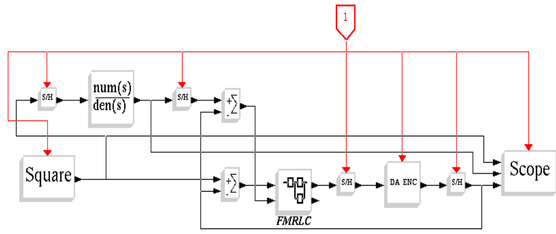

Fig. 8. Scicos schema with FMRLC

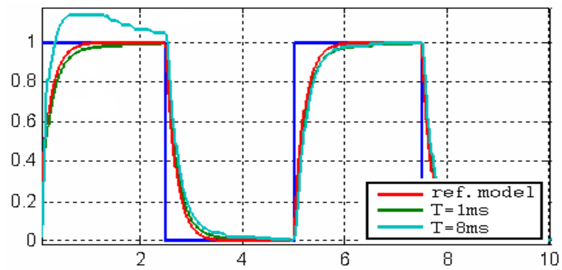

Fig. 9. Servomechanism real-time step response experiment with full FMRLC configuration

On Fig. 9 demonstrative results of the experiments are presented.

For different control periods T the servomechanism gives different results (Fig. 9). As it is shown on Fig. 9 the system response with T=1 ms is very close to the model reference response. It must be emphasized that FMRLC rapidly gains the knowledge of how to control the plant. At the beginning of the experiments all the rules consequents were set to zero.

D. DC servo control with wireless signal transmission

The third example of using Scilab/Scicos/RTAICodeGen, RTAI and RTAI-Lab for servo control is described in this section. Unlike the already described systems this one uses wireless transmission (Fig. 10) between a controller and an object.

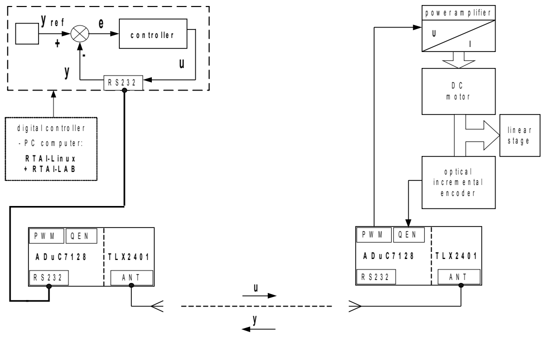

Fig. 10 . Structure of servomechanism with wireless signal transmis sion

In the system, instead of the RT-DAC4/PCI card, transmission via the RS232 interface and two modules with the ADuC7128 controllers and the TLX2401 radio modules are used. The modules make bidirectional wireless transmission possible for PWM control signal, position measurements and additional information. The structure of the communication frames can be freely arranged by the module software. Special blocks were created for Scicos, designed for a real-time communication via the RS323 interface and signal multiplexing/demultiplexing. These blocks allow the integration of the developed hardware with the RCP software environment. The real plant with transmission modules is presented on Fig. 11.

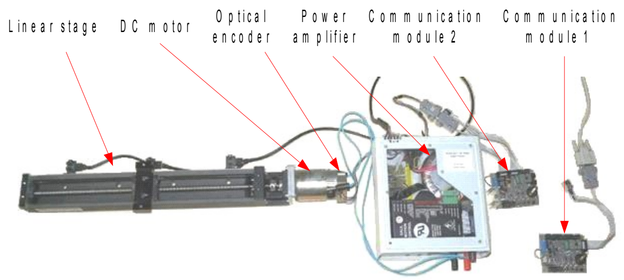

Fig. 11. Servomechanism with wireless signal transmission

The system was primarily designed as a didactic equipment used to test control algorithms in case of long delays or data loss. A few simple solutions are tested by students:

1. Keeping the last control value active until the next one arrives (zero order hold) versus resetting (setting to zero) control while data does not arrive.
2. Extension control period over many communication periods.
3. Using predictive control calculation based on object model equation on the controller side.

The example of Scicos schema for the control system with the PID discrete controller and anti-wind-up system is shown on Fig. 12. The control period for control loop is 3 ms and there is a possibility to count the number of lost data packets.

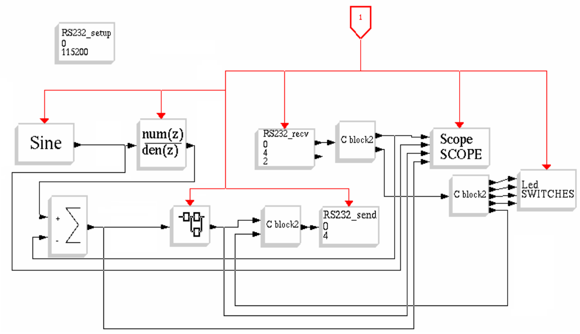

Fig. 12. Scicos schema for wireless servomechanism with PID controller

On Fig. 13 the demonstrative results of the experiments are shown. The reference model response reflects the desired response shape obtained from a simulation model without wireless communication. Remote plant responses were registered in the case of approximately 25% and 75% loss of data packets during the experiment. The level of the data loss was tuned by modifying the distance between the modules and placing barriers on waves propagation way. The degradation of responses quality according to higher number of lost packets is observed.

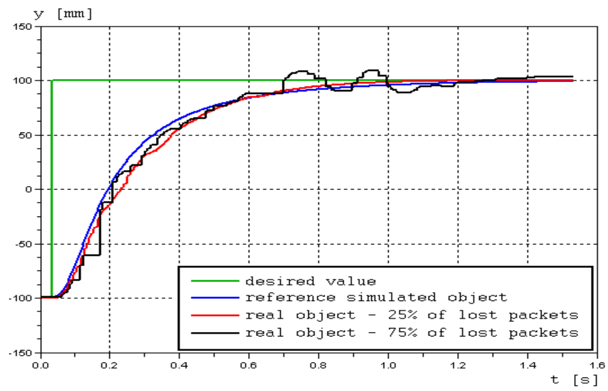

Fig. 13. Wireless servomechanism real-time step response experiment with PID controller

The remote communication module can also simulate object transfer function. There are a few advantages of this solution for students, e.g.:

1. It is possible to control objects that are not physically available in the laboratory.
2. Object model can be precisely defined and is not disrupted by unmodeled real plant dynamics. It allows students to concentrate on long delays or data loss compensation.

The transfer function given for a simulated object can be expressed in s or z domain. If the transfer function is given in s domain it is converted to z domain in the Scicos context script (automated action) applying a bilinear transformation. Finally, coefficients of the transfer function are inserted into a specially designed Scicos block (`ControlSetup`) that sends them to the remote communication module when the controller application starts. The process was automated as much as possible using the RCP environment. Only information that must be directly given by a student (object transfer function and controller diagram) is inputted manually.

Fig. 14 presents the demonstrative results of the system step response with a remote simulated object given by the transfer function $G(s)=\frac{36}{s^2-36}$.

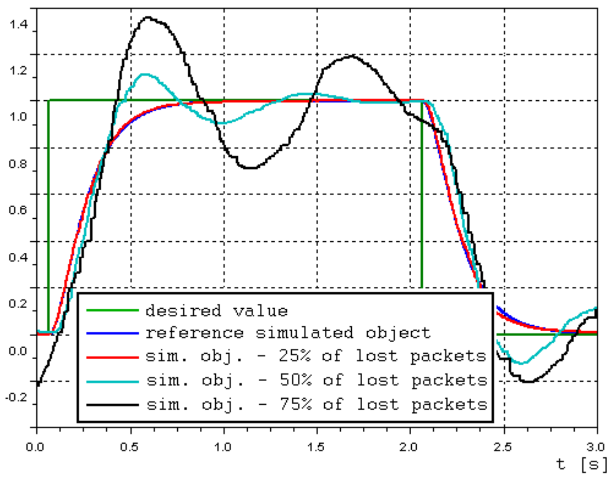

Fig. 14. The system step response with the remote simulated object

The three examples described in this section show the possibility of using Scilab/Scicos and RTAI for educational purposes. The mechatronics group in the Division of Informatics and Control at the Rzeszów University of Technology has the experience in using the Matlab/Simulink/RTW suit [15]–[17] as well as Scilab/Scicos/RTAI [4], [10]. Positive experience in using Scilab/Scicos/RTAI generated the definition of a new research goal. The goal is to build tool-chain for RCP on destination embedded hardware TS-7300 with the ARM9 processor and the FPGA chip.

IV. RCP TOOL-CHAIN IDEA FOR EMBEDDED SYSTEM

A. *Tool-chain idea*

Typical use of the Scilab/Scicos/RTAI suit involves a PC computer as a hardware platform in order to develop, run and test controllers. The goal defined in the mechatronics group in the Division of Informatics and Control assumes developing RCP tool-chain for embedded platform TS-7300. PC computer will be used as a platform for: developing controllers in Scilab/Scicos, cross-compiling controller code for embedded target, data acquisition and visualization. The executable controller code will be automatically deployed and run on the embedded platform. The GUI interface will be used on a PC computer for visualization and data acquisition. The sequence of actions taken up when tool-chain will be used is presented on Fig. 15.

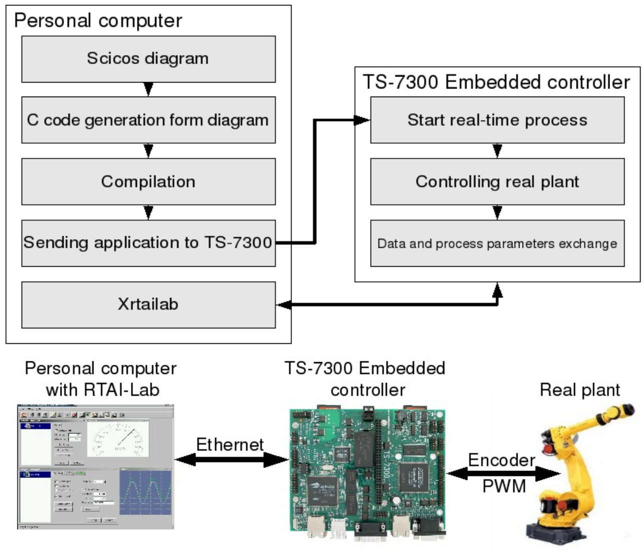

Fig. 15. Tool-chain for embedded system

In order to obtain remote GUI interface, middle-ware layer based on UDP [2] is going to be used.

To achieve the goal the following main problems are de fined to be solve:

1. Recompilation of the RTAI kernel in order to enable floating point operations in real-time tasks. TS-7300 is equipped with MaverickCrunch math co-processor.
2. Modification of automatic code generator in order to support network communication between *xrtailab* and real-time tasks without *net-rpc* mechanism which is not available for the ARM platform.
3. Developing Scicos blocks for input/output operations using encoder counters and PWM outputs implemented in the FPGA chip.

Embedded platforms are considered to be more proper for a real-time control than general PC machines. Therefore, the RCP for embedded system can be used for didactic and in dustrial purposes.

The created tool-chain will be used for developing control systems for CNC machines and laboratory robots. The CNC machine control system based on RT-Linux [18] will be exchanged with newly developed system using the tool-chain. The tool-chain will be also used for didactic purposes i n courses of Embedded Systems and Real-Time Control.

B. *The TS-7300 embedded platform*

The embedded platform consists of the TS-7300 Single Board Computer (Fig. 16) equipped with the RTAI real-time operating system.

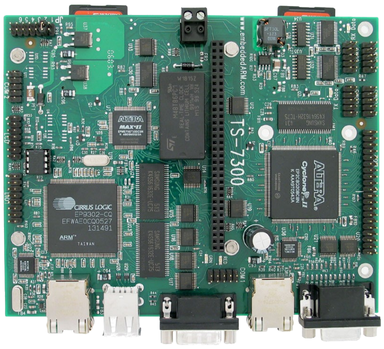

Fig. 16. TS-7300 Single Board Computer

The TS-7300 embedded system is a multipurpose board with a Cirrus Logic EP9302 200MHz processor. EP9302 features an advanced ARM920T processor designed with a memory management unit (MMU) that allows the support of high level operating systems such as Linux with a real-time extension. The modern processor connected via Wishbone, an open source hardware high-speed bus with Altera FPGA increases the flexibility of the system. The FPGA allows the development of user defined logic which can perform common controller tasks that are difficult, CPU intensive, or impossible to accomplish in software with regular DIO/GPIO hardware and the facilities of the processor. The TS-7300 provides up to 35 DIO lines connected directly to the FPGA through DIO header which can be used as a interface for control applications. In order to create servo controller, a student has to change the default FPGA configuration by implementing encoder counters and PWM generators. Peripherals such as SD Card flash socket, VGA controller, 2 USB ports, 10/100 Ethernet ports, RS232 serial ports are also included on the board.

In order to operate on the board it is possible to connect a mouse, keyboard and a monitor to USB and VGA connectors. The board can be used with additional laptop and SSH via Ethernet. For files transfer SCP software can be employed.

In order to adapt TS-7300 for servo control purposes as well as for cooperation with the Scilab/Scicos/RTAI suite the following problems were solved:

1. A cross-compiler for the embedded platform was developed. The created cross-compiler supports C, C++ and Fortran programming languages. The support for Fortran is required for automatic code generation from Scicos schemas. The cross-compiler delivered with TS-7300 does not support Fortran.
2. The RTAI kernel delivered with TS-7300 was modified in order to allow running real-time tasks in a user mode. In the original version of kernel delivered with TS-7300 a real-time task running in user mode caused system crash.
3. Encoder counters and PWM outputs have been implemented using Verilog language and Quartus II software tool in FPGA chip.

The tool-chain will let students to graphically design control structure, simulate system with plant model, deploy controller code into embedded system, run and test system for real-plant control.

V. Conclusion

Scilab/Scicos/RTAI is low cost, high quality and an open source Rapid Prototyping System. The system can be used on courses of Control Theory, Embedded Systems, Real-Time Control and Mechatronics as an alternative to Matlab/Simulink. The system can be also applied in control system development for embedded platforms with ARM processors. It seems that the number of embedded industrial applications developed using Scilab/Scicos/RTAI should rapidly grow in the nearest future. According to the above mentioned expec tation teaching students applying Scilab/Scicos/RTAI in the development of embedded control systems is indispensable.

References

[1] R. Bucher, L. Dozio, *CACSD with Linux RTAI and RTAI-Lab*, Valencia [Online], 2003, ftp://ftp.realtimelinuxfoundation.org/pub/events/rtlws-2003/proc/bucher.pdf
[2] R. Bucher, L. Dozio, *Rapid Control Prototyping with Scialb/Scicos and Linux RTAI* , http://www.scilab.org/events/scilab2004/final_paper/1 - bucher_rtai2scilab.pdf
[3] http://www.mathworks.com
[4] G. Skiba, T. Żabiński, K. Wiktorowicz, *Rapid prototyping of servo controllers in RTAI-Lab* , VI Conference on Computer Methods and Systems, 141–146, Cracow 21-23 November 2007.
[5] http://www.inria.fr
[6] http://www.enpc.fr
[7] S. L. Campbell, J. P. Chancelier, R. Nikoukhah: *Modelling and Simulation in Scilab/Scicos* , Springer 2006.
[8] http://www.rtai.org
[9] http://www.comedi.org
[10] A. Bożek, G. Skiba, T. Żabiński, *Rapid prototyping of control systems in RTAI-Lab* , X Krajowa Konferencja Robotyki (to be published), Piechowice 3–6 September 2008.
[11] M. Petko, T. Uhl, *Embedded controller design – mechatronic approach* , Proc. Of the Second International Workshop on Robot Motion and Control RoMoCo 2001, pp. 195–200.
[12] http://www.inteco.cc.pl
[13] T. Żabiński, *Tuning PID controllers for servo* , PAR, vol 4 (134), 56-63, 2008.
[14] K. M. Passino, S. Yurkowicz, *Fuzzy Control* , Addison-Wesley, 1998.
[15] D. Marchewka, T. Żabiński, *Adaptive neural controller for the task of trajectory tracking of the 3DOF manipulator* , VI Conference on Computer Methods and Systems, 195–200, Cracow 21-23 Novem ber 2007.
[16] T. Żabiński, T. Grygiel, B. Kwolek, *Design and implementation of visual feedback for an active tracking* , ICCVG, Warsaw, Machine Graphics & Vision, 2006.
[17] T. Żabiński, A. Turnau, *Compensation of friction in robotic arms and slide tables* , 16 th IFAC World Congress, Prague, July 4 to July 8 2005.
[18] T. Żabiński, *Control of mechatronics systems in real-time – classical and intelligent approach* , PhD thesis, AGH University of Science and Technology, Cracow 2006.