Implementation of the First Order Control System

ŠTEFAN CHAMRAZ
and
RICHARD BALOGH
Slovak University of Technology in Bratislava

Abstract

Lectures for the Computer Architecture are conceived to get the students acquaint with the architecture and basic properties of the computers in general with emphasis to the basic building components and their properties. We created a set of closely interconnected lessons and exercises on the embedded systems. The idea of lectures is to show an implementation issues and to show an interconnection between the theory and practice. The basic ideas of the lectures and our experiences from its application are presented. We also present the problems with the digital implementation of the continuous controller, proper choice of the sampling period and replacing the output D/A converter with a PWM signal generator. Results and the comparison of the measured parameters with theoretical ones are presented.

Mathematics Subject Classification 2000: 93-01, 93C83, 68M99
General Terms: Computer Architecture, Sampling Period
Additional Key Words and Phrases: embedded system, control system, sampling period, PI and PS controllers, RISC architecture

1. INTRODUCTION

Computer Architecture course is the part of the bachelor studies at our faculty for many years. Few years ago we adapt them significantly in order to make them appropriate for the majority of the students. Lectures are conceived to get students acquaint with the basic properties of computers in general with emphasis to basic building components and their properties. This is closely associated with possibilities of its practical use. In the higher level studies students are pointed more to the theoretical topics of the control systems analysis and synthesis, often with missing connection between the theory and practice: how to implement the control algorithms in the embedded microcomputer. Therefore we created a set of closely interconnected lessons and exercises on the embedded systems. The idea of lectures is to show an implementation of the digital PS controller based on single chip micro-controller Atmel AVR with RISC architecture [Turley 1997]. Besides to the controller algorithm itself we implemented also the algorithms for manual

Author's address: Š. Chamraz, R. Balogh, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia. mail: stefan.chamraz@stuba.sk, richard.balogh@stuba.sk
The project was supported by the grant KEGA 3/5201/07.

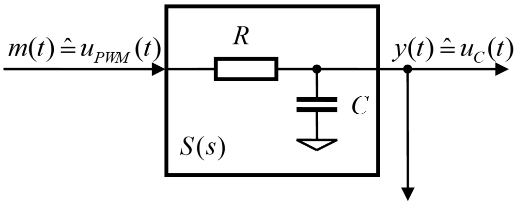


Fig. 1. Model of the controlled system.

and automatic control, local and remote parameter adjustments, event logging and control system upgrades. As the hardware base it was used the school evaluation board called MiniMexle [Pospiech et al. 2006]. Its use was mentioned also in [Balogh 2008]. See also simmilar topics in [Lodge 2002].

Presented problem should also answer the following questions for students:

—What is the reason to have different memory types and areas in the microcomputer (RAM, EEPROM, FLASH, boot section, application section, etc.)?

—What are the purposes of the interrupt subsystem, embedded timers and counters (RTC, watchdog, PWM), serial interfaces (UART, TWI, SPI)?

—How to use embedded A/D and D/A converters?

—Why do we need all of them, when the control algorithm itself requires only few registers and some mathematical operations?

In this article we will show details of the interconnection between the real A/D converter and its sampling rate. Another focus will be on the replacement of the D/A converter with the internal PWM generator. The overall concept of the lecture is to bring the students from the theoretical design and simulation of the controller to the real (and operating) application. Also we try to break the idea that digital control system is not so good as corresponded analogue circuit.

Requirements on the final application take into the account the implementation on the 8-bit single chip micro-controller:

—Range of the controlled voltage 0 − 5 V.

—Required precision better than 1%.

—Controlled voltage ripple less than 5 mV.

—Elimination of disturbances.

The last requirement will be mentioned just marginally. Major part of the theoretical works assumes simply addition of the disturbance to the system output and it does not change the system dynamics. In our system the disturbance (change of the output load) not only that changes the output voltage, but also influences the overall characteristics of the system, including its dynamic behavior.

2. THE MODEL OF THE SYSTEM

2.1 Controlled System

The controlled system is the simplest first order system created by the resistor and capacitor (see Fig. 1). Our goal is to control the capacitor voltage. At this moment we control the system without any additional load on its output.

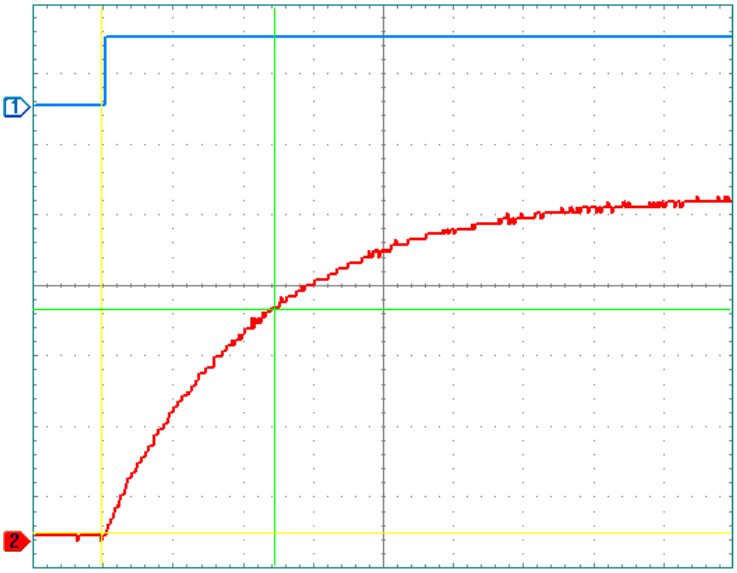


Fig. 2. Measured step response of the system. Scope CH1 (blue) 5 V/div, CH2 (red) 1 V/div, Time base 200 ms/div. Measured $T_{63} = 492.0\,\mathrm{ms}$

The values of $R = 10\,\mathrm{k}\Omega$ and $C = 50\,\mu\mathrm{F}$ were selected. For these values, corresponding *time constant* $T = RC$ is 0.5 s. Transfer function of such system is

$$S(s) = \frac{K}{sT + 1} = \frac{1}{1 + 0.5s} \tag{1}$$

We do not describe here the selection of the R and C values, but their choice is limited e.g. by the maximum output current available, maximum frequency on the inputs of the A/D converter of the micro-controller, input impedance of the A/D filter and other factors.

2.2 Parameter Identification

Implemented manual control mode was used also for verification and identification of the system. Measured data were captured using an oscilloscope (see Fig.2) and the time constant and gain were measured.

Real parameters of the system measured from the step response of the system were very close to the theoretical values.

3. CONTROLLER DESIGN

3.1 Continuous PI Controller

From the classical control theory point of view, our goal corresponds with the continuous control system with PI controller (see Fig. 3). This figure (and also the following ones) are displayed with „*physical*" quantities – figure contains also the

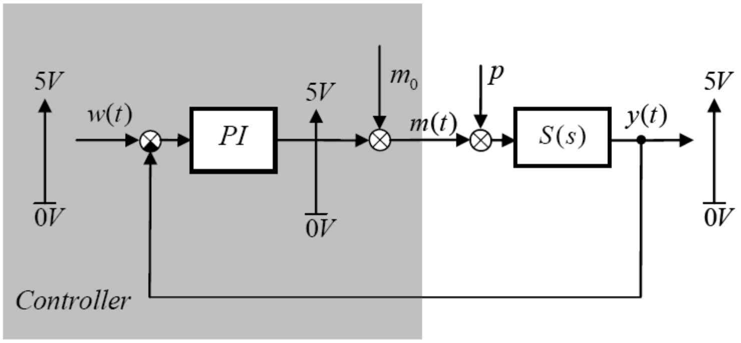


Fig. 3. Continuous control system.

ranges for inputs and outputs. The transfer function of the PI controller is

$$R(s) = K_R \left(1 + \frac{1}{T_I s}\right) \tag{2}$$

Parameters for the controller were designed using the *inverse dynamics method* as following: $K_R = 1[-]$ and $T_I = 0.5[\mathrm{s}]$.

This corresponds to the following continuous version of the controller:

$$PI: \quad m(t) = m_0 + K_R \left(e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau\right); \quad e(t) = w(t) - y(t) \tag{3}$$

3.2 Transformation to the Discrete Version

Of course we cannot implement continuous system in digital micro-controller, so we need to replace the equation (3) with a discrete one. We have to replace the integral with a sum using e.g. backward or forward rectangle method [Balátě 2004]. Of the above will result in following PS controller equation:

$$\begin{aligned} PS: \quad m(nT_v) &= m((n-1)T_v) + m_0 + K_R \left(e(nT_v) + \frac{T_v}{T_I} e((n-1)T_v)\right); \\ e(nT_v) &= w(nT_v) - y(nT_v) \end{aligned} \tag{4}$$

These equations are the minimal part of the PS controller program. They are stored in a program memory (FLASH), parameters of the controller in the permanent data memory (EEPROM). If we want to modify the control algorithm, we can use the uploader in a boot-loader area of the program memory.

The control system was implemented according to the Fig. 4, which assumes using the A/D and D/A converters. The figure emphasizes the discretization in amplitude, but it is better to redraw the figure to emphasis the discretization in time (see Fig. 5), because the digital correction system is implemented on micro-controller with Harvard architecture with glimpses of the parallelism. Micro-controller can perform pipelining, some activities can be done independently. Even with this advantages it is still a serial machine, so the processing of the control algorithm, together with utilities takes certain amount of time which is not necessary constant. The time delays influence the overall system stability. Stability is influenced by the

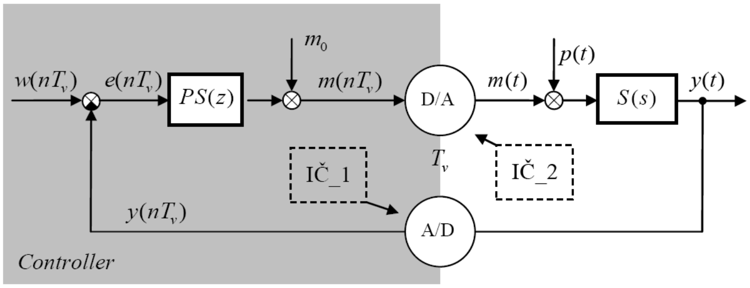


Fig. 4. Digital control system.

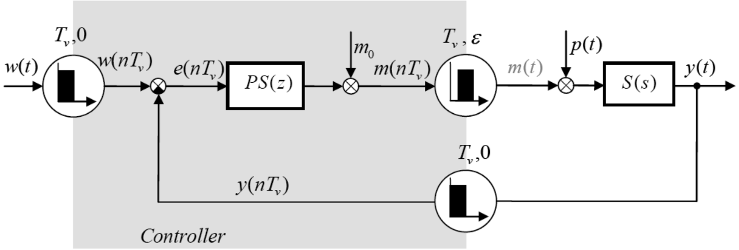


Fig. 5. Digital control system - signal delays.

sampling period and the relative shift of the output. External manifestation of both of them is the time lag.

Based on this we will assume both the discretization in amplitude and in time. We will use the pulse component with the sampling period T_v and relative shift ε.

3.3 Sampling period

We use the internal 10-bit A/D converter integrated on the AVR chip with a 5.000 V voltage reference V_{REF}. Its LSB corresponds $5\,[\mathrm{V}]/1024 = 4.88\,[\mathrm{mV}] \doteq 5\,[\mathrm{mV}]$. The value of the V_{REF} indirectly determines also the range of measured and controlled quantities. Strictly from the technical point of view, the $V_{REF} = 2.56\,\mathrm{V}$ is more appropriate, but this required more complicated hardware arrangement.

We cannot control more precise than we can measure. The maximum theoretical precision of the control is 5 mV. Taking to the account all the imperfections of the successive approximating A/D converter this is sufficient to achieve the required 1% precision.

3.3.1 *Design of the sampling period.* From the theory and literature we have many possibilities (see also [Chamraz 2008]):

a) According to the Shannon Sampling Theorem (indirectly assumes the overall gain of the open-loop system to be $KK_R = 1$) the maximum frequency is

$$\omega_R = \omega_N = \frac{1}{T} = \frac{1}{0.5[s]} = 2\ [\mathrm{rad\ s}^{-1}]$$

This value corresponds with the maximum period of the oscillations $T_R = \pi$ [s] and

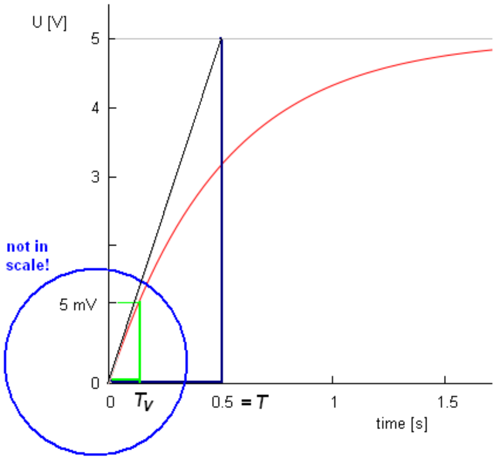


Fig. 6. Choice of the minimal sampling period.

maximum allowed sampling period $T_v = \pi/2 \doteq 1.57$ [s].

b) Empirical rule saying that T_v should be $\frac{1}{6} \div \frac{1}{10}$ of the maximum time constant (again assuming the open loop gain 1). This rule gives $T_v = \frac{T}{10} = \frac{0.5}{10} = 0.05$ [s].

c) Rule of thumb (often used in literature).

d) Now consider the discretization in amplitude. For maximum allowed change of the controlled value (5 mV) we can determine the maximum change of $y(t)$, i.e. its derivation

$$\frac{dy(t)}{dt} = \frac{5[\mathrm{V}]}{0.5[\mathrm{s}]} = 10.0\ [\mathrm{V\ s}^{-1}]$$

We can assume, that if during the sampling period T_v measured value will change less than 5 mV, we will be under the resolution of the A/D converter (see Fig. 6).

Directly from the Fig. 6 we have

$$\frac{T_v}{T} = \frac{5[\mathrm{mV}]}{5[\mathrm{V}]} \Rightarrow T_v = 0.5[\mathrm{s}]\frac{5[\mathrm{mV}]}{5[\mathrm{V}]} = 0.5[\mathrm{ms}]$$

e) Let T_w is a desired control dynamics and relative offset of the action is $\varepsilon = 1$. It is possible to show that when we design the PS controller that it's sampling period satisfies condition

$$T_v \leq 0.22 T_w. \tag{5}$$

Then the transient process will correspond to the 2nd order system with the max. overshoot less than 1% (considering the equality in (5)). The settling time will be $t_{reg} \doteq 2.84T_w + \varepsilon_v T_v$. The sampling is not necessarily synchronized with the control step. If the relative offset of the action is $\varepsilon = 0$, then the transient process will come near to the first order system and it will take *longer* time $t_{reg} \doteq 5T_w$. We can choose the control dynamics equal to the time constant of the system: $T_v \leq 0.22T_w = 0.22 \cdot 0.5 = 0.11$ [s], and it is more than the minimal value.

When comparing results of the a) and b) the ratio of these values is more than 30. In this article we use the value $T_v = 0.11$ [s] with the relative offset $\varepsilon \in <0, 1)$.

The answers to the questions ,,Why $T_v = 0,22T_w = 0,11[s]$, why it is an advantageous to choose $\varepsilon = 1$, why the PS controller algorithm has an equation (4)?" are

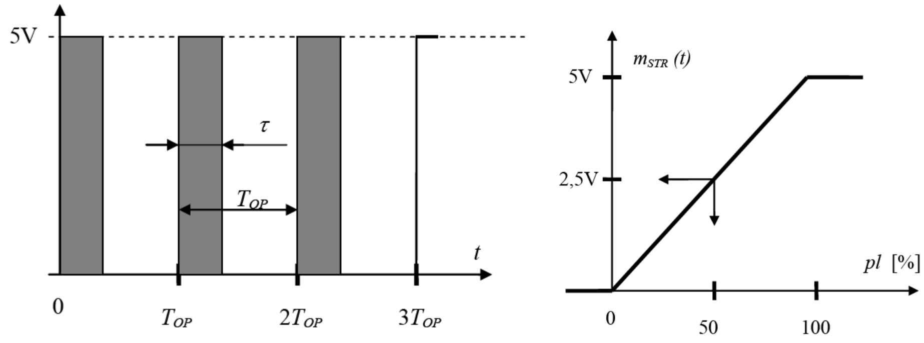


Fig. 7. a) Parameters of the PWM signal. b) PWM ideal transfer function.

beyond the limits of this article. We just outline here that controller algortihm implemented by the computer can be viewed as a spectrum converter with a time lag. Here we just use the modified results from the [Chamraz 2008], which is an answer to the students questions about the different methods of numerical integration and different methods for proper choice of the sampling period.

It should be noted that we determine the sampling period from the point of view of the control algorithm. For the conversion we set the A/D converter clocks. The conversion consists of the two basic phases – sampling (takes about 1.5 cycle), and conversion (takes 13 cycles). Sampling circuit is principally a capacitor with the capacity approx. 20 pF. This capacitor must be charged (resp. discharged) during the sampling phase through a set of resistors. This limits the maximum frequency f_{ADC} of the A/D conversion. Maximum f_{ADC} according the datasheet [Atmel 2007] is 200 kHz. This can be also used to estimate the maximum sampling frequency.

Note also that half of this frequency should not be considered as the maximum frequency which can occur at the input to the A/D converter.

Note: PS controller, transformed from the PI controller, behaves like a system with variable time lag. Controller output is "breathing". This phenomenon is called *jitter* [Martí et al. 2001]. Impact of the variable time lag we eliminated with the above limitations for T_v.

3.4 Substitution for the D/A converter: PWM

The micro-controller used in this project contains a built-in A/D converter with a multiplexer, but it lacks of a D/A converter. To manage this problem we will use the internal PWM generator instead. Pulse-width modulation (PWM) uses a square wave whose pulse width is modulated resulting in the variation of the average value of the waveform. PWM signal (see Fig.7a) has a repeating period T_{OP} and τ is the time when the output is active. During the time $(T_{OP} - \tau)$ is the output inactive (log. 0). The ratio

$$pl = \frac{\tau}{T_{OP}}, pl \in <0, 1)$$

is called *duty cycle*.

Fig. 7b shows the "ideal" transfer characteristics of the PWM block, where

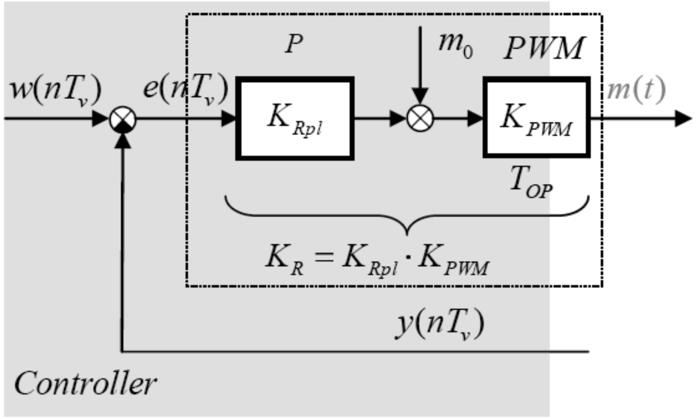


Fig. 8. Overall gain of the controller.

$m(t) = m_{STR}$ represents the median value of the PWM signal. The gain of this block is

$$K_{PWM} = \frac{5\,[\mathrm{V}]}{1\,[-]} = 5\,[\mathrm{V}]$$

Internal structure of the controller corresponding to its proportional part is shown at the Fig.8.

If we assume just P controller with gain $K_R = 1$, this value corresponds with the gain $K_{Rpl} = pl\,[-]/e\,[\mathrm{V}]$ of the value $K_{Rpl} = \frac{K_R}{K_{PWM}} = \frac{1}{5} = 0.2[\mathrm{V}^{-1}]$.

From the Fig. 7b it is clear, that PWM is a non-linear block. Its output will be limited $0 \leq m(t) < 5\,[\mathrm{V}]$.

3.5 PWM frequency

We assumed the PWM output as an analogue value. To satisfy that we need to filter out the high frequency components of the signal $m(t)$ and to remain only the DC component. That requires:

—to add a filter between the processor and a controlled system, or

—such a design of the entire circuit that system itself filters the signal.

The second approach was used, PWM signal is filtered by the system itself.

The Fig. 9 shows the PWM signal with a duty cycle of 25 % as measured directly on the processor (blue) and after (insuffucient) filtering by the system (red). The voltage ripple is 240 mV. PWM period is 131 ms, T_{ON} is 31 ms, T_{OFF} is 100 ms.

3.5.1 *Design from the frequency characteristics.* If we choose the frequency of the PWM signal $\omega_{OP} = 10T^{-1} = 20\,[\mathrm{rad.s}^{-1}]$ then oscillations with a frequency $\omega_R = 2\,[\mathrm{rad.s}^{-1}]$ will be supressed 10 times. This corresponds to the PWM period (see. Fig.7a)

$$T_{OP} = \frac{2\pi[\mathrm{rad}]}{20\,[\mathrm{rad.s}^{-1}]} \doteq 0.31\,[\mathrm{s}]$$

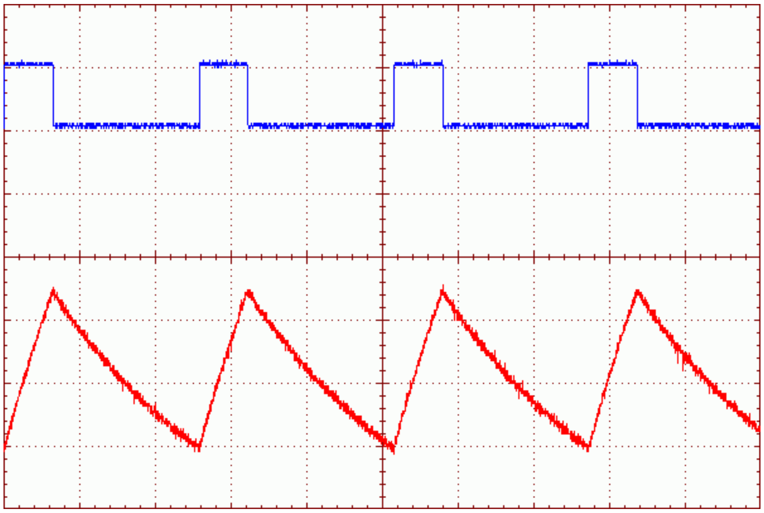


Fig. 9. 25% PWM signal on the oscilloscope. Scope: CH1 (blue): 5 V/div, CH2 (red): 100 mV/div, Time Base 50 ms/div. ATmega88 parameters: Timer 1, Mode PWM, Phase Correct, 10-bit, Prescaler 1:1024, OCR1A = 255, internal 8 MHz RC oscillator used.

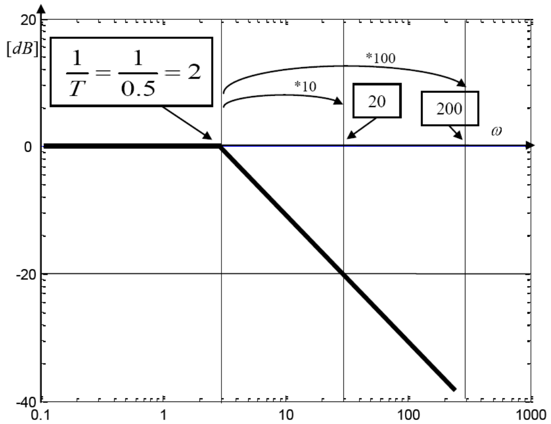


Fig. 10. Design of the PWM frequency from the frequency characteristics.

When the PWM frequency $\omega_{OP} = 200\,[\mathrm{rad.s}^{-1}]$, then oscillations with a frequency $\omega_R = 2\,[\mathrm{rad.s}^{-1}]$ will be suppressed 100 times. This corresponds to the PWM period

$$T_{OP} = \frac{2\pi[\mathrm{rad}]}{200\,[\mathrm{rad.s}^{-1}]} \doteq 0.0314\,[\mathrm{s}] \doteq 30\,[\mathrm{ms}].$$

3.5.2 *Validation using the maximum ripple of the controlled value.* Let's suppose that we use the above value $T_{OP} = 30\,[\mathrm{ms}]$. Let the PWM duty cycle is at 0.5 (see Fig.11). The theoretical steady-state value is 2.5 V. During the log.1 the signal exponentially increases up to the maximum with the time constant $T = 0.5\,[s]$. During the log. 0 it decreases with the same time constant to 0 V. The ratio $\frac{T_{OP}/2}{T} = \frac{15\,[\mathrm{ms}]}{0.5\,[\mathrm{s}]} = 30$ is relatively large number, thus we can replace exponentials

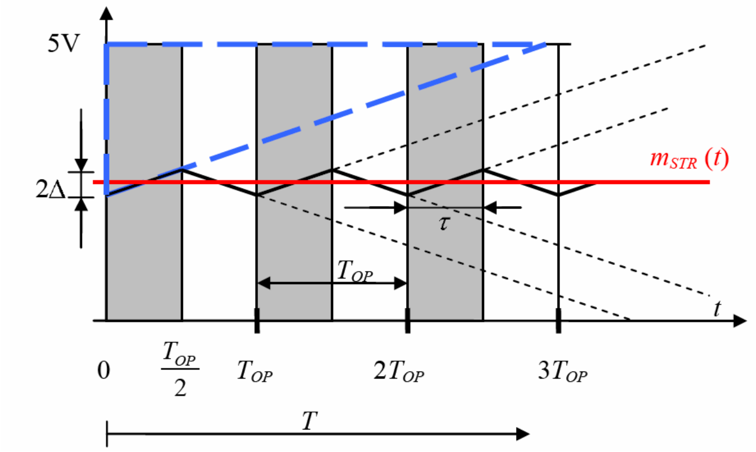


Fig. 11. Maximum ripple of the controlled value (not in scale!).

with lines.

To determine the real ripple value we state from the Fig. 11, where $V_{max} = 5\,[\mathrm{V}]$

$$\frac{V_{max} - (m_{STR}(t) - \Delta)}{2\Delta} = \frac{T}{T_{OP}/2} = \frac{0.5\,[\mathrm{s}]}{15\,[\mathrm{ms}]}$$

Its solution is $2\Delta \doteq 77\,[\mathrm{mV}]$, and this is more than 15 LSB of the A/D converter. It is necessary to redetermine the T_{OP} such that $\mathrm{LSB}/2 = 2.5\,[\mathrm{mV}]$

$$\frac{2.5\,\mathrm{V} + (\mathrm{LSB}/2)}{LSB} = \frac{0.5\,[\mathrm{s}]}{T_{OP}/2}$$

with solution for $T_{OP} \doteq 2\,[\mathrm{ms}]$. The closest possible smaller value, if possible, we set according the properties of the used timer/counter. Similarly as for the frequency of the A/D converter, we are not able to set the frequency of the PWM signal to any value, just to the certain particular values depending on the selected timer mode and oscillator and pre-scaler settings.

4. EVALUATION AND EXPERIMENTS

The above theoretical analysis was first verified using the Matlab/Simulink. The model of the system, analogue PI and discrete PS controllers were created. Quantization in amplitude was also assumed. The PWM block was simulated in 8-bit mode.

From the Figures 3, 4, 5 it is clear that disturbances are easily simulated on the input or output of the system. Theoretically it is possible to make the step change of the disturbances at the input of the system, but practical implementation would probably be difficult. We will move closer towards to the reality and rather simulate the disturbance as a change of the output load. Parallel to the capacitor, the resistor of the value 20 kΩ is connected. Note that this also changes the gain and the time constant of the controlled system.

Control algorithm was written in C language, programmed into the microcomputer and tested. Real controller has added funcionality compared to the MATLAB

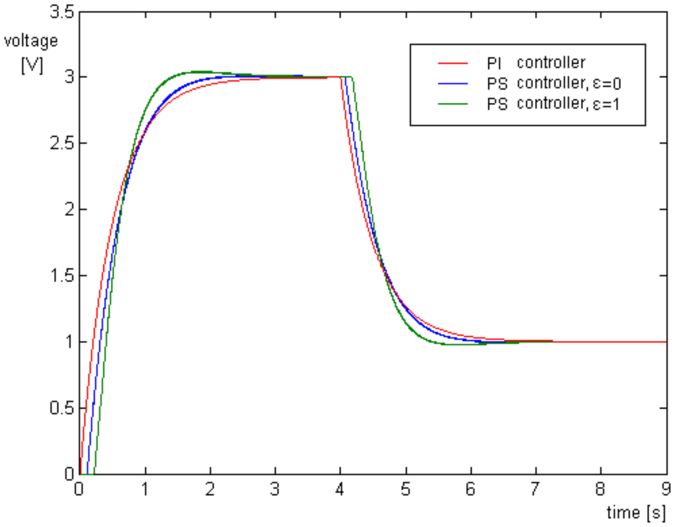


Fig. 12. Simulation results - step change of the setpoint. Red: PI continuous controller ($K_R = 1$ and $T_I = 0.5$ [s]), blue: PS controller transformed from PI, $\varepsilon = 0$, green: PS controller transformed from PI, $\varepsilon = 1$.

simulation. We added functions like the Anti-reset WindUp (ARW), manual mode etc. All the calculations were performed in an integer arithmetic. The results are very close (identical) to the results of the simulations in MATLAB.

4.1 Design of the Controller for step change of the setpoint

Parameters of the PS controller were set to the following values: $K_R = 1\,[-]$, $T_I = 0.5\,[\mathrm{s}]$, $T_v \doteq 0.11\,[\mathrm{s}]$, $\varepsilon = 1\,[-]$, $T_{OP} = 2\,[\mathrm{ms}]$. Fig. 12 shows the results for the step change of the setpoint value from 0 to 3 V and later back to 1 V.

4.2 Design of the Controller for step change of the disturbance

Because the controller is designed for the setpoint step change, its properties are not optimized also for changes of the disturbance. If there is a requirement for the proper operation also at changing disturbances, it is necessary to design the controller with *two degrees of freedom* to increase the loop dynamics. This can be achieved with increasing the proportional gain to $K_R = 3\,[-]$ and decreasing the sampling period $T_v \doteq 0.037\,[\mathrm{s}]$.

Simulation on Fig. 13 shows increase to the value of 3 V without additional load. At the time $t = 3\,\mathrm{s}$, the parallel $20\,\mathrm{k}\Omega$ resistor was added at the output and at the time $t = 6\,\mathrm{s}$ it was again removed. Full line correspondents to the settings for the setpoint step change, dotted line for changes of the load (disturbances).

5. CONCLUSION

The concept was tested during the laboratory exercises on the CAD of Control Systems lectures at the FEI STU during the winter semester in the 2008. The system

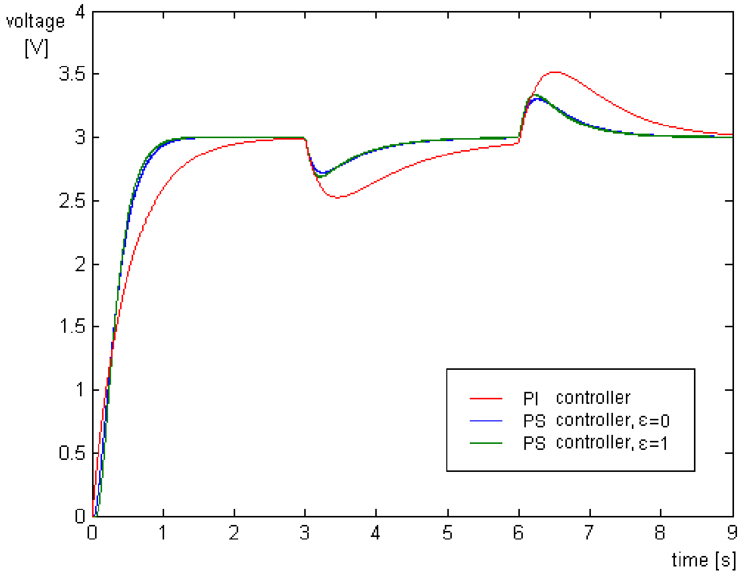


Fig. 13. Simulation results - step change of the setpoint and disturbances. Red: PI continuous controller ($K_R = 1$ and $T_I = 0.5$ [s]), blue: PS controller designed for disturbance, $K_R = 3, T_v = 0.037$ [s] and $\varepsilon = 0$, green: PS controller designed for disturbance, $K_R = 3, T_v = 0.037$ [s] and $\varepsilon = 1$.

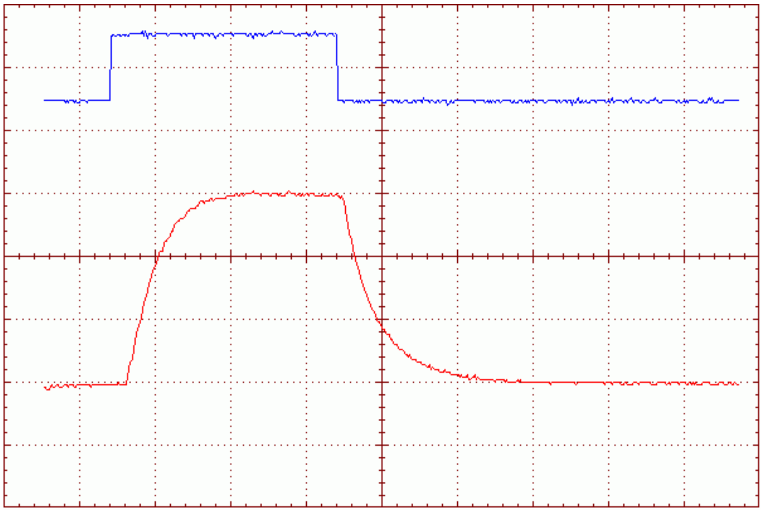
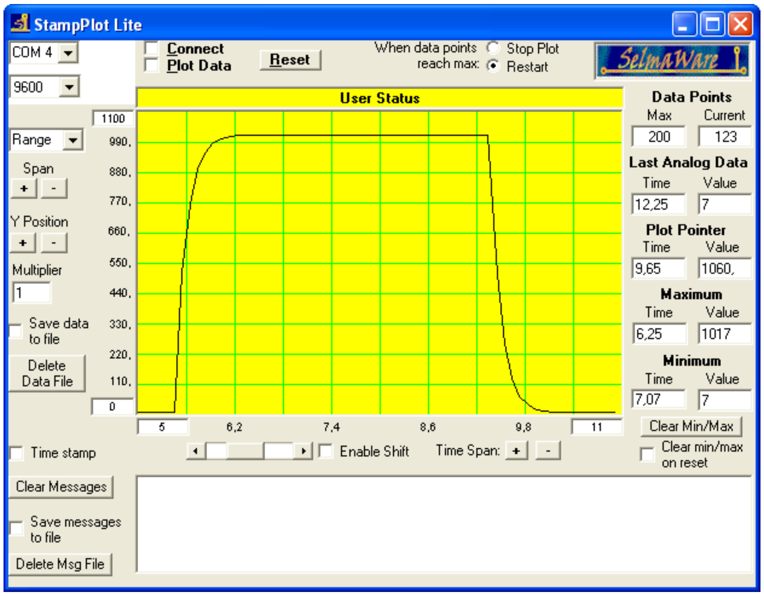


Fig. 14. Control process as measured on the oscilloscope. Scope: CH1 (blue): 5 V/div, CH2 (red): 1 V/div, Time Base 1 s/div. ATmega88 with internal RC oscillator 8 MHz used. First step is from 1 V to 4 V ($\varepsilon = 1$), second one from 4 V to 1 V ($\varepsilon = 0$).



StampPlot Lite

COM 4

9600

□ Connect

□ Plot Data

When data points reach max: ○ Stop Plot ● Restart

Range

Multiplier 1

□ Save data to file

□ Time stamp

□ Enable Shift

□ Save messages to file

□ Clear min/max on reset

	Data Points	
	Max	Current
	200	123
Last Analog Data	Time 12,25	Value 7
Plot Pointer	Time 9,65	Value 1060,
Maximum	Time 6,25	Value 1017
Minimum	Time 7,07	Value 7

Fig. 15. Student's measurement on the system.

was implemented on the evaluation board MiniMexle with the Atmel ATmega88 processor. We used the AVR Studio with GNU avr-gcc compiler, and programs were written in C language. For visualisation we use the StampPlot software [Hebel and Devenport 2006], [Hernandez 2009], which can plot diagrams on-line, from the data arrived from the serial communication interface. The students were not very excited, considered the presented topic too difficult, without any visible connection to another lectures they already passed.

From the simulation it is clear that the controller with two degrees of freedom has better quality properties. The improvements were achieved by the changes of the sampling period and the gain of the controller. This should motivate students to study more consistent and lead to the recognition of the fact that everything can be calculated. Even the sampling period. If the calculated values are correctly implemented in microcomputer we obtain satisfactory results.

All the study materials for the students are available on-line [Balogh 2009].

ACKNOWLEDGMENTS

The project was supported by the grant KEGA 3/5201/07.

REFERENCES

Atmel. 2007. *ATmega48/88/168. 8-bit Microcontroller with 8K Bytes In-System Programmable Flash*, Rev. 2545MAVR09/07 ed. Atmel Corporation, San Jose, USA.

Baláte̋, J. 2004. *Automatické řízení*, 2 ed. BEN - technická literatura, Praha. ISBN 80-7300-148-9.

Balogh, R. 2008. Basic activities with the Boe-Bot mobile robot. In *DidInfo 2008. 14th International Conference.* FPV UMB, Banská Bystrica, Slovakia.

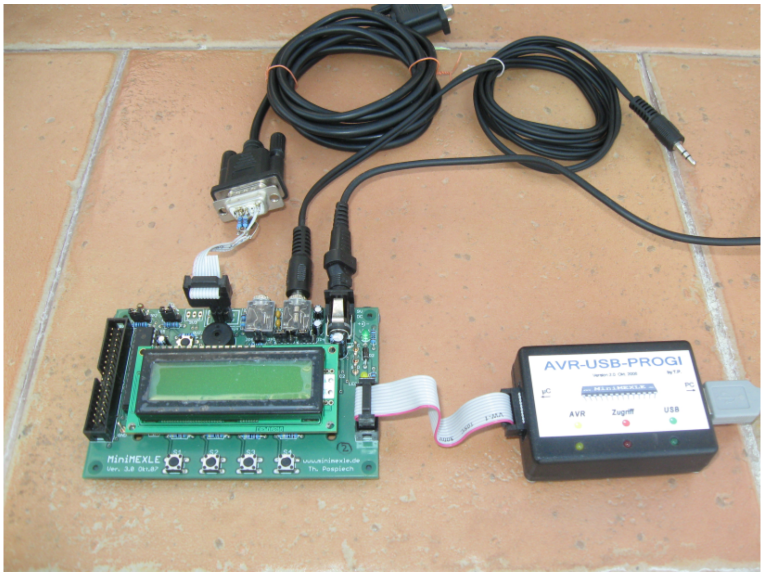


Fig. 16. Evaluation board with implemented system.

Balogh, R. 2009. CAD riadiacich systémov (http://ap.urpi.fei.stuba.sk/cadrs) – the webpage with the study materials for students.

Chamraz, v. 2008. Voľba periódy vzorkovania. (Choice of Sampling Period) [In Slovak]. In *In:Cybernetics and Informatics. International Conference. Ždiar, Slovakia*. SSKI, STU, Bratislava. ISBN 978-80-227-2828-7.

Hebel, M. and Devenport, W. 2006. *Process Control*. Parallax Inc., Rocklin, CA.

Hernandez, O. J. 2009. Using the C Stamp with the StampPlot graphing utility. Tech. rep., A-WIT Technologies Inc.

Lodge, K. B. 2002. The programming of a microcontroller as an integral part of process control for undergraduate chemical engineers. In *North Midwest Section Annual Conference of the American Society for Engineering Education*. University of Wisconsin-Madison College of Engineering, Madison, Wisconsin, U.S.A.

Martí, P., Fuertes, J. M., and Fohler, G. 2001. Sampling jitter compensation in real-time control systems. In *In Work-in-progress Session, 13th Euromicro Conference on Real-Time Systems (ECRTS01)*. Delft, The Netherlands.

Pospiech, T., Knot, J., and Gruhler, G. 2006. MiniMEXLE - the microprocessor development board for eyeryone. In *Radioelektronika - 16th International Czech-Slovak Scientific Conference*. http://www.mexle.net.

Turley, J. 1997. Atmel AVR brings RISC to 8-Bit world. *Microprocessor Report 11,* 9 (July), 3, MicroDesign Resources.

Received July 2009; accepted July 2009