

Implementation of Full-Featured PID Regulator in Microcontrollers

O. Krejcar

*Department of Measurement and Control, FEECS, VSB Technical University of Ostrava,
17. listopadu 15, Ostrava, 70833, Czech Republic, phone: +420737882422, e-mail: ondrej.krejcar@remoteworld.net,
Department of Information Technologies, FIM, University of Hradec Kralove,
Rokitanskeho 62, Hradec Kralove, 500 03, Czech Republic*

I. Spicka, R. Frischer

*Department of Automation and Computing in Metallurgy, FMME, VSB Technical University of Ostrava,
17. listopadu 15, Ostrava, 70833, Czech Republic, phone: +420597324576, e-mail: ivo.spicka@vsb.cz*

crossref <http://dx.doi.org/10.5755/j01.eee.113.7.617>

Introduction

The PID controller algorithm involves three parameters: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the action to the current error, the integral value determines the action as the sum of recent errors, and the derivative value determines the action based on the rate at which the error has been changing. The sum of these three actions known as action value adjust the system state via a action element such as a control valve[1, 2].

Note that the use of the PID algorithm does not guarantee optimal control strategy of the system or system stability.

Some applications may require using only one or two modes, a PID controller will be called a PI, PD, P or I controller.

Note: Due to the diversity of the field of control theory and application, many naming conventions for the relevant variables are in common use.

A familiar example of a control loop is mixing hot and cold water by using corresponding valves to maintain the water at the desired temperature. This typically involves the mixing of two streams, the hot and cold water. The person tests the water or measures its temperature. Based on this knowledge - feedback they perform a control action to adjust the hot and cold water valves until the process temperature stabilizes at the desired value.

In the absence of disturbances, pure proportional control will not settle at its target value, but will retain a steady state error (droop) that is a function of the proportional gain and the process gain. Only the drift component (long-term average, zero-frequency component) of process gain matters for the droop – regular or random fluctuations above or below the drift cancel out [1].

Despite the droop, both tuning theory and industrial

practice indicate that it is the proportional term that should contribute the bulk of the output change.

Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain and added to the controller output. The magnitude of the contribution of the integral term to the overall control action is determined by the integral gain, K_i .

The integral term (when added to the proportional term) accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a proportional only controller.

The rate of change of the process error is calculated by determining the slope of the error over time (i.e., its first derivative with respect to time) and multiplying this rate of change by the derivative gain K_d . The magnitude of the contribution of the derivative term (sometimes called rate) to the overall control action is termed the derivative gain, K_d .

The derivative term slows the rate of change of the controller output and this effect is most noticeable close to the controller setpoint. Hence, derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability.

Modifications too of the PID algorithm

The basic PID algorithm presents some challenges in control applications that have been addressed by minor modifications to the PID form.

Integral windup

One common problem resulting from the ideal PID implementations is an integral windup, where a large

change in setpoint occurs (say a positive change) and the integral terms accumulates a significant error during the rise (windup), thus overshooting and continuing to increase as this accumulated error is unwound. This problem can be addressed by:

1. Initializing the controller integral to a desired value;
2. Increasing the setpoint in a suitable ramp;
3. Disabling the integral function until the PV has entered the controllable region;
4. Limiting the time period over which the integral error is calculated;
5. Preventing the integral term from accumulating above or below pre-determined bounds;
6. Freezing the integral function in case of disturbances.

Many PID loops control a mechanical device (for example, a valve). The rate of mechanical wear is mainly a function of how often a device is activated to make a change. Where wear is a significant concern, the PID loop may have an output deadband to reduce the frequency of activation of the output (valve).

The proportional and derivative terms can produce excessive movement in the output when a system is subjected to an instantaneous step increase in the error, such as a large setpoint change. In the case of the derivative term, this is due to taking the derivative of the error, which is very large in the case of an instantaneous step change. As a result, some PID algorithms incorporate the following modifications:

In this case the PID controller measures the derivative of the output quantity, rather than the derivative of the error. The output is always continuous (i.e., never has a step change). For this to be effective, the derivative of the output must have the same sign as the derivative of the error.

In this modification, the setpoint is gradually moved from its old value to a newly specified value using a linear or first order differential ramp function. This avoids the discontinuity present in a simple step change.

Setpoint weighting uses different multipliers for the error depending on which element of the controller it is used in. The error in the integral term must be the true control error to avoid steady-state control errors. This affects the controller's setpoint response. These parameters do not affect the response to load disturbances and measurement noise [2–4].

Noise in derivative

A problem with the Derivative term is that small amounts of measurement or process noise can cause large amounts of change in the output. It is often helpful to filter the measurements with a low-pass filter in order to remove higher-frequency noise components. However, low-pass filtering and derivative control can cancel each other out, so reducing noise by instrumentation means is a much better choice. Alternatively, a nonlinear median filter may be used, which improves the filtering efficiency and practical performance [1, 5].

In the absence of disturbances, pure proportional control will not settle at its target value, but will retain a

steady state error (droop) that is a function of the proportional gain and the process gain. Only the drift component (long-term average, zero-frequency component) of process gain matters for the droop – regular or random fluctuations above or below the drift cancel out.

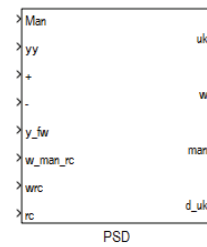


Fig. 1. Overall controller's inputs and outputs

Fig. 1 describe I/O configuration of one block of the controller. I/O ports have following means. Man port is the input of manual automat switch signal. In a manual mode the controller puts on its output the manual setting value of the plant set point. This manual set point can be set using plus (+) or minus (-) input. When one of these ports is set to nonzero value, the man set point value increases or decreases by plus or minus value for the first time slowly and next rapidly. If w_man_rc is set to one than the manual set point is set to the value of the wrc port. In automatic control mode, the port man is set to zero, plus and minus ports serve for established set point and controller according its parameters and selected control mode sets its outputs uk or d_uk. Very important (for cascade control for example) is y_fw input port. To this can be connected feed forward control signal. This feed forward can improve the quality of regulation. The w output gives back the actual value of the set point. The man output gives the value of the manual set point. Inner structure represents Fig. 2.

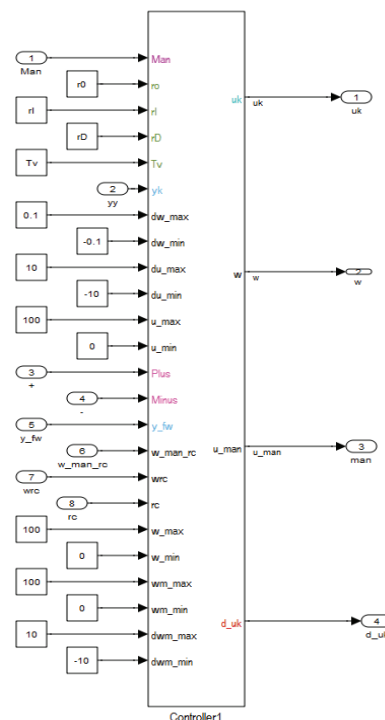


Fig. 2. All settings constants for the controller

Here we can see individual blocks. It is very important to timing counting operations. This timing signal coming-out from block called "STEP CONTROLLER". These blocks have to work in interactive mode. Variable conversion block recalculating classic PID values into values useable in incremental controller form. The Anti windup block serve to prevent step changes when "Demand values" change occur. The bump less switch ensure unharmed switch of controller from manual into automatic control.

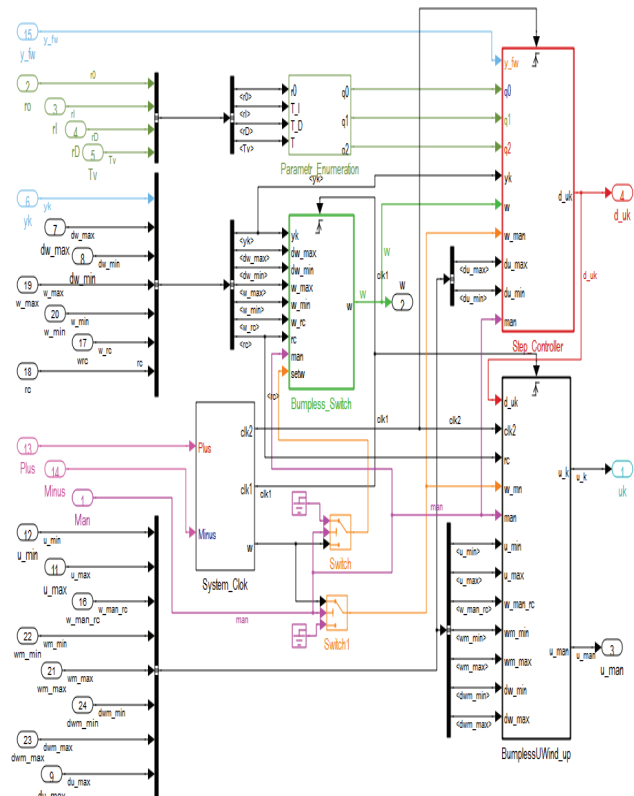


Fig. 3. Controller's inner block diagram

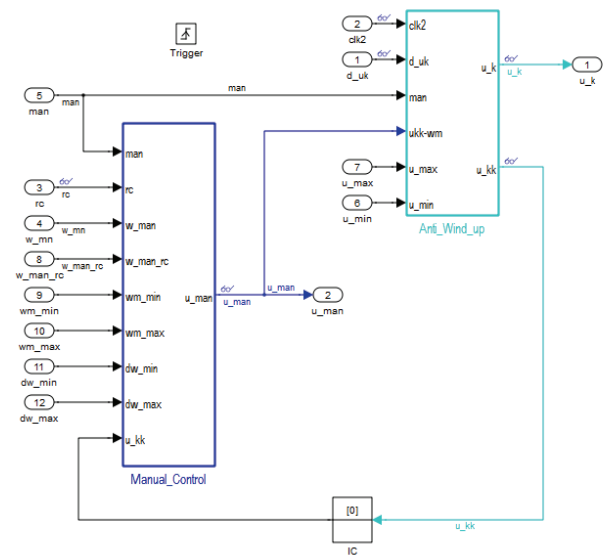


Fig. 4. The bumbles switch control

This block consist of two subsystems – anti wind up block and manual control block (see Fig. 4) Introduced

concept is making basic controller's body. It is possible to add some other features like demand value limitation, derivative part filtration and last but not least implementation of Takahashi's controller form, where control loop participate on proportional signal size only [6–11].

Simulation results

For proper regulation function is important smooth behavior. For positional PID controller realization is presented conception sufficient. But, if we intend use an incremental controller, then glimmer on differential regulators output can has unpleasant consequences on controlled system. This complaint can be fixed by disconnecting the memory in current step controller. Controller behavior without attend can be seen on Fig. 9. and behavior with attended state on Fig. 10.

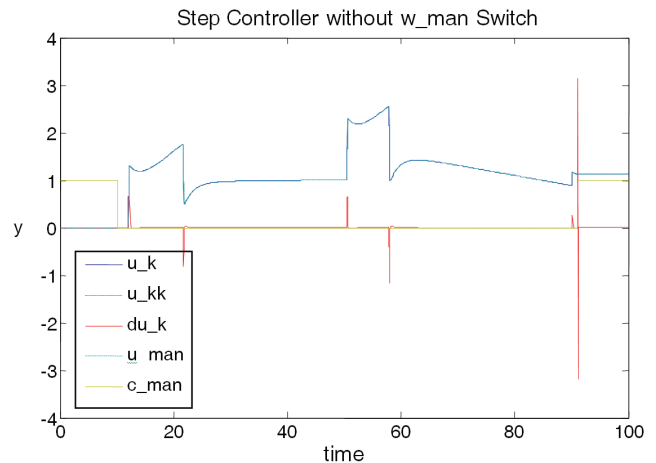


Fig. 5. Step controller without man switch

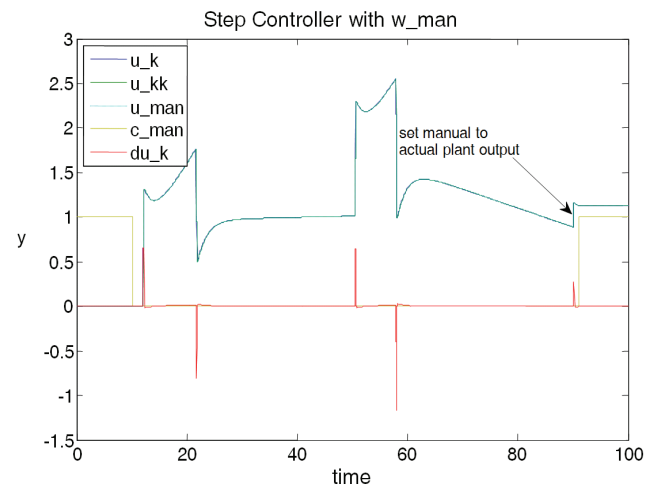


Fig. 6. Step controller with switching manual and internal value of controller

The signal u_k corresponds with output port no. 1, u_{kk} is the value of u_k signal in the previous scan. Both signals are a difference output of the controller, output of the step controller. The u_{man} (output port no. 2) is the actual value of manual set point output of the controller. The c_{man} is the signal switching the controller from the

automatic mode to the remote automatic mode that means set point is put form higher level control system (Fig. 3).

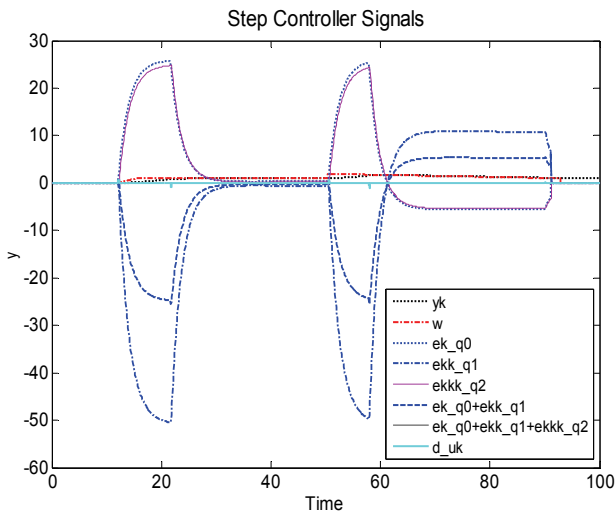


Fig. 7. Step controller block inner signal time values

Signal curves of stepper controller (Fig. 8) has following meaning: y_k is output of the plant, w is demand value, d_uk is increment of actuating quantity in given step, ek_q0 is value of actual deviation times by parameter $q0$, ek_q1 is deviation value from previous step times by $q1$ parameter and ek_q2 is deviation value from before last step times by $q2$ parameter. There are stated sums of calculated signals $ek_q0+ek_q1+ek_q2$.

To test the controller control signals man for switching from manual to automatic mode, rc for switching from autonomous to remote control and w as the set point value.

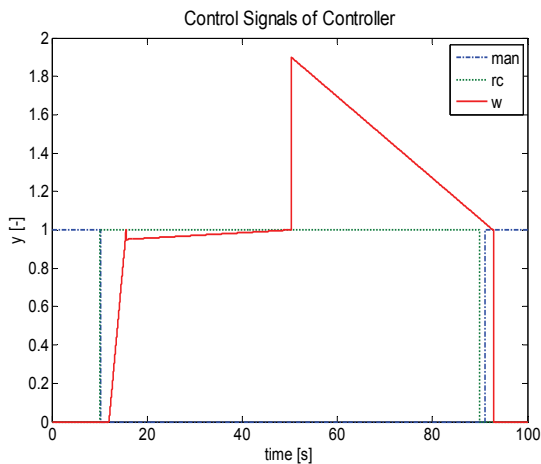


Fig. 8. Time values of setpoint w and control signals rc a man

On Fig. 9 are presented signals from closed loop control circuit. Controller is set with regard to optimal control quality. Signal y is output of regulated system, u_k is control quantity, d_uk is increment of control quantity in particular controlling steps. It is obvious, that control circuit did not show any skips of demand quantity from manual to automatic mode transition. Course of demand value reply for typical behavior of program regulation. It is obvious that designed concept is fully

conform to exacting applications where digital controller made its work.

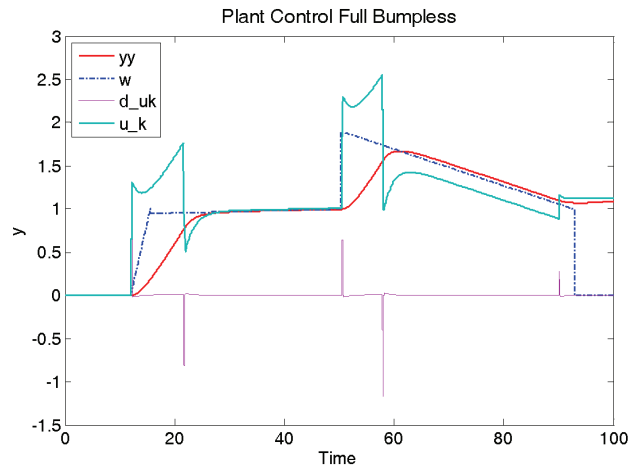


Fig. 9. Closed loop signal of plant and controller

HW part

The heart of the system is 16-bit single chip microprocessor (MCU) by MICROCHIP company with type symbol PIC24FJ64GB106 (Fig. 7). It is powerful microprocessor with many integrated interfaces and remappable pins. This MCU connect to old series of 8-bit processors but offers much more extend instruction suite and many other improvements. Previous PSD controller suffer from some inadequacies resulting from restrictions of simple, 8-bit MCU. We are minimized these issues thanks to power of the unit (CPU) and also with hardware support for multiplying and dividing.

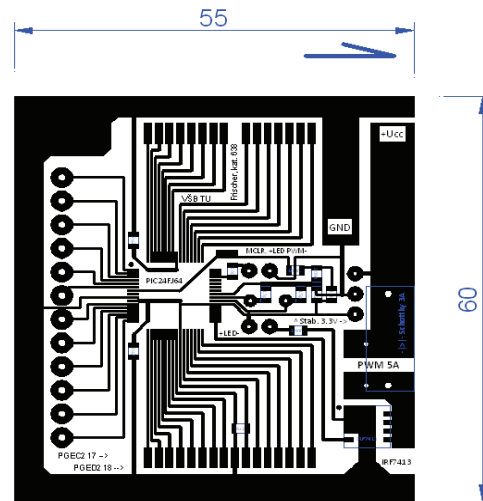


Fig. 10. Controllers PCB design with I/O interface

MCU PIC24FJ64GB106

This is very advanced, 16-bit MCU with modified Harvard architecture disposing with about 16MIPS power with clock rate of 32MHz. It's contain PLL, A Phase Lock Loop frequency multiplier, available to the external oscillator modes and which allows clock speeds of up to 32MHz. With PLL we can achieve full speed even with internal, 8MHz oscillator. The instruction's executed time

is reduced to 125ns compared to 200ns of previous model. Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP. We can do difficult computing operations thanks to native support for multiplying and dividing. All of these operations has a minimum affect to a total program execution time (compared to 8-bit version, where were necessary to design software tools for dividing and multiplying). Supply voltage is no critical, suggested 3.3V is used with withdrawal of about 15mA. As a part of the chip is 32kHz secondary oscillator which is used for RTCC (Real-Time Clock/Calendar). Main benefits are integrated two-way USB communication interface eventually native support for SPI and IIC serial communication. Another useful component of the MCU are five, independent 16-bit timers for time-critical applications (e.g. presented PSD controller). The analog variables can be read by integrated 10-bit A/D converter with 16 multiplexed inputs with 500kps sampling rate. Along with remappable pins, big amount of RAM (16K) and instructions set optimized for using C language we can use this MCU as a powerful tool for real-time applications like modern PSD controllers are. Great advantage are 29 remappable pins. The PCB design is then quite simple. Whole controller make one-sided PCB with exposed system's pins. Controller has 8 analog and 8 digital inputs. The action interventions can be generated by 8 digital outputs and with one (five max) power PWM output. a PWM native support of this MCU make it best for direct control of AC or DC motors or servo mechanisms. A PWM output server as driver for power N-MOS-FET transistor IRF7413. Thanks to its low on-state resistance, $R_{DS(on)}=0.011\Omega$ is possible to drive bigger motors without power switch cooling. If we drive 5A load, the total power dissipation on power transistor is only about 0.28W. A maximum allowed current load is 13A continuously, if we keep blocking voltage up to 30V. On Fig. 8 we can observe the PCB including text notes.

Conclusions

The article had to task to show basic mathematical describe of incremental controller and suggest structure of single blocks so that be able to implement into single chip microcontroller. Individual blocks are analyzed as far as elementary operation's level. From hardware description resulting, that selected structure is suitable for implement into similar MCU especially if they are able to conFig. I/O pins as with ours MCU.

Acknowledgements

This research was supported in part by (1) „Centre for Applied Cybernetics“, Ministry of Education of the Czech Republic under project 1M0567, (2) „SMEW – Smart Environments at Workplaces“, Grant Agency of the Czech Republic, GACR P403/10/1310, (3) „SCADA system for control and monitoring of processes in Real Time“, Technology Agency of the Czech Republic, TACR, TA01010632 and (4) "User Adaptive Systems", VSB - Technical University of Ostrava under project SP/2011/22.

References

1. **Ang K. H., Chong G. C. Y., Li Y.** PID control system analysis, design, and technology // *IEEE Trans Control Systems Tech*, 2005. – No. 13(4). – P. 559–576.
2. **Bennett S.** Nicholas Minorsky and the automatic steering of ships // *IEEE Control Systems Magazine*, 1984. – No. 4(4). – P. 10–15.
3. **Li Y., Ang K. H., Chong G. C. Y.** PID control system analysis and design – Problems, remedies, and future directions // *IEEE Control Systems Magazine*, 2006. – No. 26(1). – P. 32–41.
4. **Yang T.** Architectures of Computational Verb Controllers: Towards a New Paradigm of Intelligent Control // *International Journal of Computational Cognition*. – Yang's Scientific Press, 2005. – No. 3(2). – P. 74–101.
5. **Krejcar O.** Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering // *EURASIP Journal on Wireless Communications and Networking*. – Hindawi, 2009. – Vol. 2009. Article ID 802523 (8 p.). DOI 10.1155/2009/802523.
6. **Krejcar O., Frischer R.** Detection of Internal Defects of Material on the Basis of Performance Spectral Density Analysis // *Journal of Vibroengineering*. – Kaunas, 2010. – Vol. 12. – No. 4. – P. 541–551.
7. **Krejcar O., Frischer R.** Inexpensive Discrete PSD Controller with PWM Power Output // *Przegląd Elektrotechniczny*, 2011. – Vol. 87. – No. 1. – P. 296–299.
8. **Krejcar O., Frischer R.** Non Destructive Defects Detection by Performance Spectral Density Analysis // *Sensors – MDPI Basel*, 2011. – Vol. 11. – No. 3. – P. 2334–2346.
9. **Brida P., Machaj J., Duha J.** A Novel Optimizing Algorithm for DV based Positioning Methods in ad hoc Networks // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 1(97). – P. 33–38.
10. **Fuksis R., Greitans M., Hermanis E.** Motion analysis and remote control system using pyroelectric infrared sensors // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2008. – No. 6(86). – P. 69–72.
11. **Zlosnikas V., Baskys A.** PID controller with enhanced disturbance rejection // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2008. – No. 5(85). – P. 65–68.

Received 2011 02 14

O. Krejcar, I. Spicka, R. Frischer. Implementation of Full-Featured PID Regulator in Microcontrollers // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2011. – No. 7(113). – P. 77–82.

The goal of article is to provide a basic mathematical description of incremental controller and suggest structure of single blocks so that be able to implement into single chip microcontroller. Individual blocks are analyzed as far as elementary operation's level. From hardware description resulting, that selected structure is suitable for implement into similar MCU especially if they are able to conFig. I/O pins as with ours MCU. The article attends to the implementation of a full PID controller in the selected microprocessor. In order that PID controller met all demands of the modern control systems, it must include more than just the basic PID algorithm. Especially necessary is to implement treatment changes of the desired value by checking of its gradient. It should also be addressed treatment

changes of action variable. There can be two ways how to do that: (a) limitations of the action control variable onto the allowable interval and treatment of the permissible gradient of the action variable, (b) establish direct forward coupling for the implementation of cascade control. For many applications it is necessary to allow the controller has to be able to switch from automatic to manual mode and vice versa bumpless. Splitting of the regulator into main blocks further can enable minimized time demands of the algorithm and to limit negative effects when input changes occur. A suitable controller design methodology as a reentrant routine and defining memory areas, then save processors program memory. The proposal is based on the PID model created in the MATLAB simulation environment. Il. 10, bibl. 11 (in English; abstracts in English and Lithuanian).

O. Krejcar, I. Spicka, R. Frischer. PID regulatoriaus realizavimas mikrovaldiklyje // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 7(113). – P. 77–82.

Analizuojami pavienių valdiklių matematiniai modeliai, kuriuos siekiama sujungti į bendrą grandinę ir realizuoti kaip mikrovaldiklį. Analizuojami atskiri blokai, gebantys atlikti elementarias funkcijas. Pasirinktą PID reguliatorių galima realizuoti mikrovaldiklyje, jei įvesties ir išvesties išvadai suderinami su mikrovaldiklio išvadais. Siekiant, kad PID reguliatorius atitiktų visus šiuolaikinės valdymo sistemos poreikius, jis turi remtis ne vien PID algoritmu, bet turėti ir papildomų galimybių. Siūlomas PID modelis sudarytas programų paketo „Matlab“ terpėje. Il. 10, bibl. 11 (anglų kalba; santraukos anglų ir lietuvių k.).