TECHNICAL TRANSACTIONS 5/2018
CZASOPISMO TECHNICZNE 5/2018
ELECTRICAL ENGINEERING
DOI: 10.4467/2353737XCT.18.077.8559
SUBMISSION OF THE FINAL VERSION: 04/05/2018

Volodymyr Samotyy (vsamotyy@pk.edu.pl)
Sergii Telenyk
Department of Automatic Control and Information Technology, Faculty of Electrical Engineering, Cracow University of Technology

Petro Kravets
Volodymyr Shymkovych
Taras Posvistak
Department of Automation and Control in Technical Systems, National Technical University of Ukraine

A real time control system for balancing a ball on a platform with FPGA parallel implementation

System balansowy kulkowy w czasie rzeczywistym z równoległymi obliczeniami FPGA

Abstract

In this paper, a new PID-regulator based solution to the scientific and practical problem of increasing the accuracy of regulating the position of a ball on a platform in real time is proposed. A transfer function for balancing a ball on a platform is developed. A PID-regulator for balancing a ball on a platform is synthesized. A PID-regulator implementation on FPGA with parallel calculations is designed. An increased accuracy of regulating the position of a ball on a platform is approved by natural simulation.

Keywords: PID controller, FPGA, VHDL, real-time systems

Streszczenie

W artykule uzyskano nowe rozwiązanie rzeczywistego naukowego i praktycznego problemu zwiększania dokładności regulacji pozycji kulkowych na platformie w czasie rzeczywistym za pomocą regulatorów PID. Przy ich realizacji sprzętowej na FPGA z równoległymi obliczeniami przeprowadzono model matematyczny obiektu sterującego oraz opracowano sprzętowy komponent dla FPGA.

Słowa kluczowe: regulator PID, FPGA, VHDL, systemy czasu rzeczywistego

1. Introduction

In our post-industrial era, the need for automation has not diminished. The use of the Internet of things, driverless cars, robots and other complex objects with many parameters define the scope of new challenges for control systems in terms of their development and efficiency. Therefore, the development, research and design of real-time control systems by complex objects with many parameters are an actual task.

It is important to note that studies of follow-up systems are relevant. This is due to the use of autopilots and other systems that interact with common objects. An essential characteristic of such systems is unpredictability of their critical states. Then this object can't defined by clean mathematical definitions for example by differential equations [1–4].

With the above point in mind, it is important to investigate and search for the most effective systems that can be implemented for tasks involving regulation as well as to increase the number of parallel-operating regulators. This, in turn, positively affects the speed and the power of the system. Such possibilities appear in the hardware realisation of components of control systems on Field-Programmable Gate Array (FPGA) [5–8]. FPGA has a few important characteristics that distinguish them from ordinary microcontrollers. At first, FPGA have higher computing power. At second, FPGA provides the ability to change the configuration for control computing tasks. At third, FPGA provides real parallel computing because of the programs are implemented on hardware level and do not compete for resources. At means of one FPGA chip the big count of controllers can be implemented to control a many parameters at same time at a very height speed.

2. Formulation of the problem

Fig. 1. A general view of the motion-control platform

The purpose of this work is to develop and research the approach to effective implementation of FPGA controllers for objects and processes control with many parameters. This approach will be realised for the task of balancing a ball on a platform. An additional goal of the implementation is to demonstrate the ability of FPGA controllers to simultaneously implement the image recognition system, Proportional Integral Derivative (PID) controller in real time.

To test the developed control systems, a stand was set up in the form of a square supplemented by two servo motors in the workshop [9–10]. Above the area of the square, a camera was installed to fix the parameters relating to placement and movement of the object on the field. The purpose of the regulation is to bring the object to a state of equilibrium at a certain point, working out the movement along a long given trajectory. (Fig. 1).

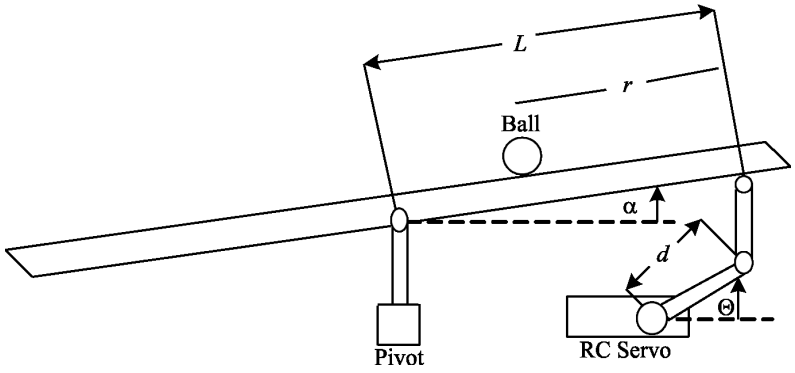

Fig. 2. Physical modelling of the ball movement on the platform

Based on the symmetry of the platform on which the control object is located, it can be argued that the formulas of the regulator along the *X* axis and along the *Y* axis will be the same. Therefore, it is appropriate to develop a mathematical model for only one axis. A simplified image of physical modelling of the ball movement on the platform with the plane and actuators is shown in Fig. 2.

3. The problem solution and experimental results

To simplify the calculations, let's suppose that slip and friction when moving the ball are absent. Let's consider the Lagrange equation:

$$\left(\frac{J}{R^2}+m\right)\ddot{r}+mg\sin\alpha-mr\dot{\alpha}^2=0 \tag{1}$$

where:

- R – the radius of the ball (0.015 m);
- m – the mass of the ball (0.01 kg),
- d – the extension length of the servomotor (0.05 m);
- g – the acceleration of free fall;
- L – the distance from the end to the middle of the platform (0.35 m);
- J – the moment of inertia of the ball ($9.99 \cdot 10^{-6}$ kg·m^2);
- r – the distance from the centre of the ball to the edge of the platform;
- α – the angle of rotation of the plate;
- Θ – the angle of rotation of the servomotor.

Therefore, it seems appropriate to create an adequate mathematical model for one axis, and then implement this model in each of the two controllers, designed to control the movement of the ball on its axis. If we use near-zero angles of the platform position, we can assume that, in the case of a Tailor series, value of function sin α for angle α will be equal to value of this angle α. There is a similar situation for function cos α. For near-zero angle α value of cos α is equal 1. This suggests taking into account the near-zero angle α and near-zero speed of the platform. The third

component in left part of equation (1) can be neglected (for example, if the speed is 0.1 radians per second, then the square of the speed gives a very small contribution 0.12 = 0.01). Therefore:

$$\alpha \approx 0 \rightarrow \begin{cases} \sin(\alpha) \approx \alpha \\ \cos(\alpha) \approx 1 \\ (\dot{\alpha})^2 \approx 0 \end{cases}$$

Therefore the equation (1) can be rewritten as:

$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\alpha \tag{2}$$

Taking into consideration the construction of platform, the angle α of this platform rotation can be approximated by means of equation (3):

$$\alpha = \frac{d}{L}\theta \tag{3}$$

In this equation: d – the length of the servomotor lever; L – the half of platform side (platform is a quadrate 40 sm × 40 sm, the side is equal 40 sm).

We substitute equation (3) into equation (2) and obtain equation (4):

$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\frac{d}{L}\theta \tag{4}$$

Now we have grounds to proceed to create a transfer function through the Laplace transform using equation (4):

$$\left(\frac{J}{R^2} + m\right)R(s)s^2 = -mg\frac{d}{L}\theta(s) \tag{5}$$

Based on the features of the control object, it is expedient to transfer the transmission function through the ratio of the ball position $R(s)$ and servomotor rotation angle $\theta(s)$:

$$T(s) = \frac{R(s)}{\theta(s)} = -\frac{mg\dfrac{d}{L}}{\left(\dfrac{J}{R^2} + m\right)s^2}\left(\frac{m}{rad}\right) \tag{6}$$

From equation (6), by substituting the corresponding data we obtain a transfer function:

$$T(s) = \frac{0.9482}{s^2}.$$

The received transfer function $T(s)$ is simulated in the Matlab Simulink package and the dependence of the ball position from the servomotor rotation angle is obtained. The structural scheme of control system of the ball on the platform in the Matlab Simulink software package is shown in Fig. 3.

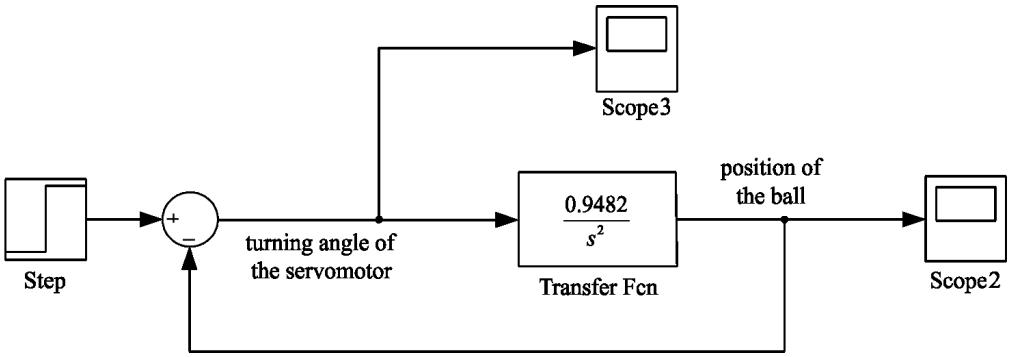

Fig. 3. The structural scheme of control system of the ball position on a platform

Let us consider the results of simulation. Fig. 4a shows the dependence of the ball position from time. Similarly Fig. 4b shows the dependence of the servomotor rotation angle from the time. When the servomotor is rotated an angle around 0.17, the ball moves from one side of the platform to the other.

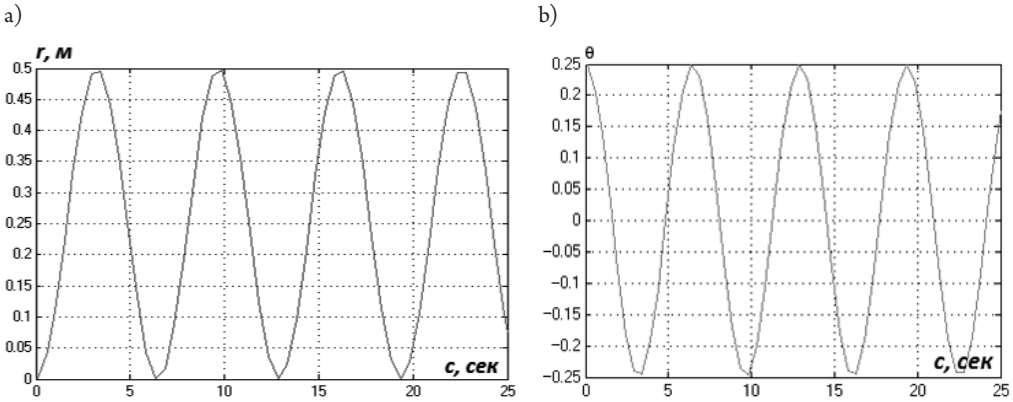

Fig. 4. The control simulation results of the ball motion on a platform: a) dependence of the ball position from time, b) dependence of the servomotor rotation angle from the time

To calculate the stabilisation of the ball movement in dependence from rotation angle, the ball control system on the platform with the PID controller was simulated in the Matlab Simulink package. The coefficients of the PID controller (the proportional coefficient K_p, the integral coefficient K_i and the derivative coefficient K_d) were calculated with the use of the special PID regulator block. Calculated values of these coefficients were K_p = 1.1636, K_i = 0.01 and K_d = 1.3626. When modelling the positioning control system with the PID controller, expected results were obtained. These results are shown in Fig. 5.

In this work, for the implementation of a closed-loop control system, an algorithm for PID for hardware implementation is used. This algorithm belongs to one of the most commonly used classes of algorithms in the theory of control. The results obtained in the work prove that it can be effectively used to control the servomotor.

a) b)

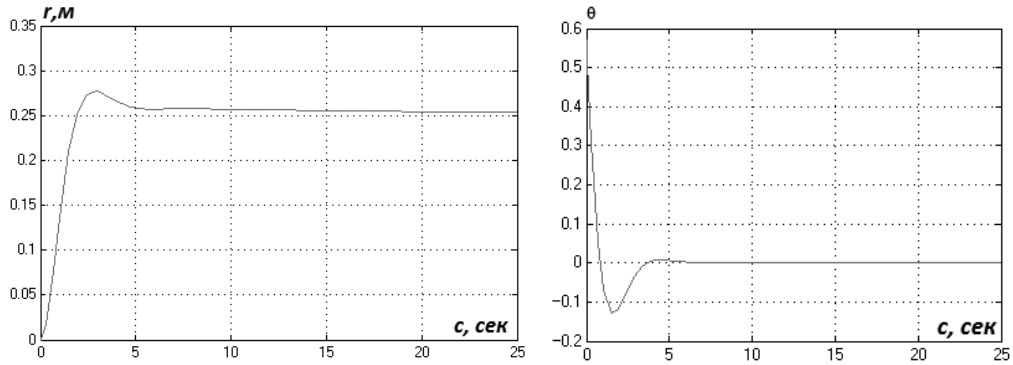

Fig. 5. The control simulation results of the ball motion on a platform with the PID controller: a) dependence of the ball position from time, b) dependence of the servomotor rotation angle from the time

The PID controller is described in differential equations:

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right] \tag{7}$$

where:

K_p – the proportional coefficient,
T_i – the integral time constant,
T_d – the derivative time constant.

For a small sampling interval T, this equation can be turned into a differential equation by discretisation. A differential equation can be implemented by a digital system, either in hardware or software. The derivative term is simply replaced by a first-order differential expression and the integral can be calculated through the sum; thus, the differential equation is given as:

$$u(n) = K_p\left[e(n) + \frac{T}{T_i}\sum_{j=0}^{n} e(j) + \frac{T_d}{T}\big(e(n) - e(n-1)\big)\right] \tag{8}$$

Equation (8) can be rewritten as:

$$u(n) = K_p e(n) + K_i\sum_{j=0}^{n} e(j) + K_d\big(e(n) - e(n-1)\big) \tag{9}$$

where:

$K_i = K_p T/T_i$ – the integral coefficient,
$K_d = K_p T_d/T$ – the derivative coefficient.

To compute the sum, all past errors, $e(0)...e(n)$, have to be stored. This algorithm is called the 'position algorithm' [2]. An alternative recursive algorithm is characterised by the calculation of the control output $u(n)$ based on $u(n-1)$ and the correction term $\Delta u(n)$. To derive the recursive algorithm, at first let's calculate $u(n-1)$ taking into consideration equation (9):

$$u(n-1) = K_p e(n-1) + K_i \sum_{j=0}^{n-1} e(j) + K_d \left(e(n-1) - e(n-1) \right) \tag{10}$$

Then the correction term can be calculated as:

$$\Delta u(n) = u(n) - u(n-1) = K_0 e(n) + K_1 e(n-1) + K_2 e(n-2) \tag{11}$$

where:

$$\begin{aligned} K_0 &= K_p + K_i + K_d \\ K_1 &= -K_p - 2K_d \\ K_2 &= K_d \end{aligned} \tag{12}$$

Equation (11) is called the 'incremental algorithm'. The current control output is calculated as:

$$u(n) = u(n-1) - \Delta u(n) = u(n-1) + K_0 e(n) + K_1 e(n-1) + K_2 e(n-2) \tag{13}$$

For the hardware implementation of the PID controller, the incremental algorithm (Eq. 13) can avoid the accumulation of all past errors $e(n)$ and can realise smooth switching from manual to automatic operation, in contrast to the position algorithm.

In the PID controller increasing the proportional gain K_p can increase the system response speed, and it can decrease the steady state error, but not eliminate it completely. Additionally, the performance of the closed-loop system becomes more oscillatory and takes longer to settle down after being disturbed as the amplification factor increases. To avoid these difficulties, the integral control K_i and the derivative control K_d can eliminate the steady-state error and improve the stability of the system.

A one-channel design was constructed based on the PID control algorithm. The PID incremental algorithm (Eq. 13) is decomposed into its basic arithmetic operations:

$$\begin{aligned} e(n) &= P_d + (-P), \\ p_0 &= K_0 \cdot e(n), \\ p_1 &= K_1 \cdot e(n-1), \\ p_2 &= K_2 \cdot e(n-2), \\ s_1 &= p_0 + p_1, \\ s_2 &= p_2 + u(n-1), \\ u(n) &= s_1 + s_2. \end{aligned} \tag{14}$$

For a parallel design, each basic operation has its own hardware unit – or adder or multipliers, which are the main elements of combinational logic. For serial design, which is composed of sequential logic, all operations share only one adder and one multiplier.

The equation (12) was used to calculate the coefficients:

$$K_0 = 1.1636 + 0.01 + 1.3626 = 2.5362$$
$$K_1 = -1.1636 - 2 \cdot 1.3626 = -3.8888$$
$$K_2 = 1.3626$$

The computational process implemented by this algorithm and the hardware implementation of the PID controller in parallel calculations on a FPGA took 284 look-up tables. The speed, as the total delay of the combinational circuit of the block, was 45.1 ns – this is 30% faster than consistently calculating all operations. Two parallel PID regulators on the same chip, which control the platform by the axes of OX and OY, used 568 look-up tables. The synthesis and simulation was executed in Xilinx ISE Design Suite 13.2 and ISE Simulator (ISim), using the Spartan 3 chip family. When using chips of other families, the rates of occupied resources and performance will be different. The modelling of the ball motion control on the platform with the high speed parallel PID controller on a FPGA is shown in Fig. 6.

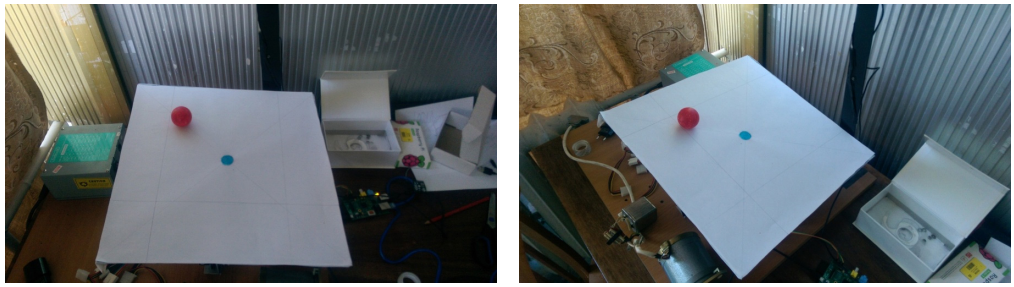

Fig. 6. The modelling of the ball motion control on the platform with the high speed parallel PID controller on a FPGA

4. Conclusion

In this paper, a new solution to the scientific and practical task of increasing the efficiency of regulating the ball position on a platform in real time has been presented. This solution uses PID-regulators with their implementation unit on a FPGA with parallel computations. A two parallel PID controller on one FPGA chip has been developed to control of two parameters of one object independently of each other. Parallel processing of the contour calculations of the PID controller that control the object has been ensured, the output of the regulator has been normalised. An increased accuracy of regulating the ball position on a platform is approved by natural simulation. The designed controllers can be used for real time controlling high-speed objects with many controlled parameters.

References

[1] Trimeche A., Sakly A., Mtibaa A., Benrejeb M., *PID Controller Using FPGA Technology,* Advances in PID Control, edited by Valery D. Yurkevich, September 6, 2011.

[2] Somani A., Kokate R., *Realization of FPGA Based PID Controller for Speed Control of DC Motor Using Xilinx SysGen,* [in:] Satapathy S., Joshi A. (eds.) Information and Communication Technology for Intelligent Systems (ICTIS 2017), Vol. 2, ICTIS 2017, Smart Innovation, Systems and Technologies, Vol. 84., 2018, Springer, Cham, DOI 10.1007/978-3-319-63645-0_9

[3] Aboelaze M., Shehata M.G., *Implementation of multiple PID controllers on FPGA,* IEEE International Conference on Electronics, Circuits, and Systems (ICECS) 6–9 Dec. 2015, Cairo, Egypt, 2015, 446–449.

[4] Krasňanský R., Dvorščák B., Kozák S., *Hardware Realization of Embedded Control Algorithm on FPGA,* COMPUTATION TOOLS 2014, The Fifth International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, 2014, 13–18.

[5] Wei Zhao, Byung Hwa Kim, Larson A.C, Voyles R.M., *FPGA Implementation of Closed-Loop Control System for Small-Scale Robot,* ICAR '05, Proceedings 12th International Conference on Advanced Robotics, 2005*, 70–77*.

[6] Chang Y.F., Moallem M., Wang W., *Efficient implementation of PID control algorithm using FPGA technology,* Proceedings of 43 IEEE Conference on Decision and Control, vol. 5, Dec. 2004, 4885–4890.

[7] Krasňanský R., Dvorščák B., *Design and Implementation of FPGA-based PID controller*, [in:] ACCS'13: 3rd International Conference on Advanced Control Circuits and Systems, ERI, Luxor, Dec. 2013, 43.

[8] Aboelaze M., Shehata M.G., *Implementation of multiple PID controllers on FPGA,* Electronics Circuits and Systems (ICECS) 2015 IEEE International Conference, 2015, 446–449.

[9] Kravets P.I., Shymkovych V.M., Fedorchuk V.V., Goy A.A., *Neural controller stability of moving object with the hardware and software realization on FPGA,*Visnyk NTUU 'KPI' Informatics, operation and computer systems, №63, 2015, 4–11.

[10] Kravets P.I., Shymkovych V.M., Samotyy V., *Method and technology of synthesis of neural network models of object control with their hardware implementation on FPGA,* 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest 2017, 947-951, DOI: 10.1109/IDAACS.2017.8095226.

If you want to quote this article, its proper bibliographic entry is as follow: Samotyy V., Telenyk S., Kravets P., Shymkovych V., Posvistak T., *A real time control system for balancing a ball on a platform with FPGA parallel implementation,* Technical Transactions, Vol. 5/2018, pp. 109–118.