International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:7, 2014

Lego Mindstorms as a Simulation of Robotic Systems

Miroslav Popelka, Jakub Nožička

***Abstract*—**In this paper we deal with using Lego Mindstorms in simulation of robotic systems with respect to cost reduction. Lego Mindstorms kit contains broad variety of hardware components which are required to simulate, program and test the robotics systems in practice. Algorithm programming went in development environment supplied together with Lego kit as in programming language C# as well. Algorithm following the line, which we dealt with in this paper, uses theoretical findings from area of controlling circuits. PID controller has been chosen as controlling circuit whose individual components were experimentally adjusted for optimal motion of robot tracking the line. Data which are determined to process by algorithm are collected by sensors which scan the interface between black and white surfaces followed by robot. Based on discovered facts Lego Mindstorms can be considered for low-cost and capable kit to simulate real robotics systems.

***Keywords*—**LEGO Mindstorms, PID controller, low-cost robotics systems, line follower, sensors, programming language C#, EV3 Home Edition Software.

I. Introduction

THE goal of the paper is to prove that Lego Mindstorms belongs to low-cost tool by that is possible to simulate, programme and test robotics systems especially for industrial compounds. Lego Mindstorms in version EV3, by which we simulated robotics systems, contains except building components of robot even large amount of sensors and detectors which enable assembled robot to orientate in space. These are mainly sensors to scan colour, sensors of distance or sensor of pressure. For our purposes we assembled robot following the line and it uses colour sensors scanning surface colour. Light sensors are used very often for these applications to scan emissivity of reflected light from surface only. In order to be able to demonstrate universality of this kit so we have decided to use PID controller being used in overwhelming majority of industrial applications. We use PID controller in our application to smooth oscillating motions to sides during following edges of black surface and transition on white surface and back. Producer of Lego Mindstorms supplies directly development environment for programming and testing robotics systems development - LEGO MINDSTORM EV3 Home Edition, which works with programmable blocks. The whole application was programmed in this development environment and it confirms fact that Lego Mindstorms is low-cost tool capable of simulating, programming and testing robotics systems for industrial compounds. Of course, programming language C# or C++ is more suitable to use in bigger and more complex projects [6], [7], [11].

Miroslav Popelka is with the Department of Automation and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlin, Nad Stranemi 4511, 760 05 Zlin, Czech Republic (phone: +420732137908; e-mail: popelka@fai.utb.cz).

Jakub Nozicka is with the Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics, Tomas Bata University in Zlin, Nad Stranemi 4511, 760 05 Zlin, Czech Republic (phone: +420 723 131668; e-mail: nozicka@fai.utb.cz).

II. Lego Mindstorms HW description

There are several sets and versions of product Mindstorms for simulation of robotics systems with possible extension by various kits. For our purposes we used kit Mindstorms, version EV3 45544, providing in its basic version big amount of components, engines, sensors, detectors for simulations, programming and testing of robotics systems in practice. Our kit uses microcomputer to control system as in real robotics systems as well. There is so called programmable box in which is placed processor, operation memory and flash memory and Linux operating system modified by producer which controls micro-computer. All our work is based on capability to use the kit as low-cost alternative to create, programme and test robotics systems in practice, later we present by creating PID controller to control robot – it is controlled by algorithm and sensors following the line. Lego Mindstorms already contains all hardware used in our work [7].

III. Description of Development Environment

Assembled robotics system may be programmed in few ways. Producer of Lego Mindstorms supplies even development environment to programme and test robotics systems together with kit. Another way to programme robot is to use programming language C# or other complex programming language. Development environment supplied with robot provides the same options as complex programming language which proves our developed application to control robot with PID controller. This solution again pointed out at universality of kit with aim to cost cutting as simulation of robotics system. It is possible to programme practically any application in development environment supplied by producer, however from point of development of more complex programs and time saving is more beneficial to use some of supported programming language. One of suitable sophisticated programming languages is C#. Company Monobrick created library for this programming language to assure communication with programming unit of robot. This library calls Monobrick Communication Library and it supports operation systems Windows, Linux and even Mac OS. To develop the application is possible to use commercial development environment Visual Studio or freely available Framework MonoDevelop / Xamarin Studio [6], [11].

IV. PID Controller

The basic idea of PID controller is to collect information

from sensor and follow-up determination of process value. Sum of proportional, integrating and derivation component represents its size. It follows to the need of back-coupling on input in order to compare with required value and to determine new process value [1]-[3].

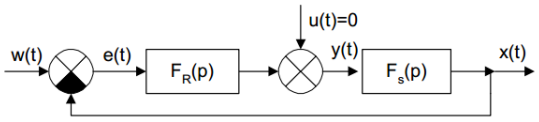


Fig. 1 A block diagram of a PID controller in a feedback loop

In general, PID controllers are being used especially in closed loop systems, see Fig. 1, in which enters two variables. First of them is setpoint w(t) and second one is disturbance value u(t). Of course, second of them is unwanted and there was effort about its reduction in constructional design [2], [3].

A. Basic features of Regulation Systems

In connection with controlling circuits is necessary to mention their basic features:

1. System Stability

Determines whether regulated system stabilizes in definite time or not. Based on roots of an equation it determines the system stability, see Fig. 2 [1]-[3].

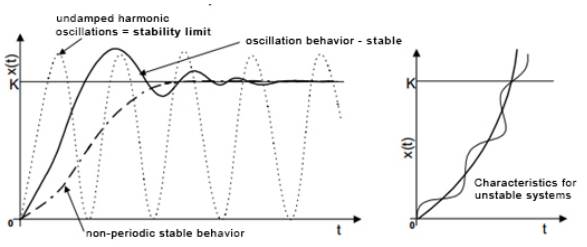


Fig. 2 Example of stable system transfer function

2. Control Accuracy and Closeness of Controlling

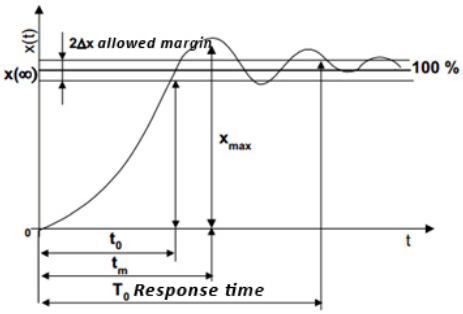
Match of controlled variable x(t) and setpoint w(t) is used to determine control accuracy, see equations 1 and 2. In case of closeness of controllingis point of interest its size of control error in steady state during error effects [1], [2].

$$e(\infty) = \lim_{p\to 0} pW(p)\, F_e(p) \qquad (1)$$

$$\Delta x(\infty) = \lim_{p\to 0} pU(p)\, F_u(p) \qquad (2)$$

3. Control Performance Quality

Is feature of controlling circuit which informs about the way and length of changed controlled variable from one value to second one. The most used way to measure regulation is step response. In Fig. 3 we can find out important information leading to determine total quality [2], [4].



In practice, it chooses usually Δx = 1 % (quality RO), 2 % and even 5 % (common RO).

We determine: t_m period of first overshoot
t_0 regulation time
Δx_{max} size of first overshoot, i.e. size of overlap (15 to 30% of fianl value $x(\infty)$)
T_0 response time $T_0 < 3\, t_m$

Fig. 3 Example of transfer function and their parameters

B. Methods of PID Controller Components Adjusting

In practice it uses the most often to estimate of PID controller parameters these empirical rules.

- adjusting by trial-and-error method
- Ziegler-Nichols method [4]

V. Adjusting PID Controller by Trial-and-Error Method

In time of using this procedure it appeared following one as the best one regarding achieved results. The procedure is based on adjusting proportional component and follow-up determination of integrating and derivation components by pre-known formula [5], [10].

1. We delete integrating and derivation components from control procedure.
2. We adjust of proportional component on value 0.5.
3. We increase the value of proportional component up to time before we receive required results regarding balance of limiting factors (response speed of system and amplitude size).
4. By reconnecting of integrating component into control procedure we achieve reduction of oscillations. The value of integrating component is adjusted on high value and during the adjustment value decreases. Decreasing of value proceeds until time when oscillation value will be suitable for concrete application.
5. Adjusting of rate response proceeds in similar way as in case of integrating component. Initial value of rate response is adjusted on value approaching zero and in the process of regulation it is being increased [2], [5], [10].

A. PID Controller Components

PID controller components divide on emerged controlled deviation that is based on reaction of controlled system. The influence each of those components on general behaviour of PID controller is specified by three controlling parameters. All components sum up and their result represents action which is commensurable to measured error. Choice of used components depends on characteristics of chosen controlled

system [1], [2], [5].

1. Proportional Component

Process value is determined at proportional component as multiple K of measured controlled deviation e(t), related to:

$$u(t) = K \cdot e(t) \tag{3}$$

when using only proportional component, so P-controller, it is coming two various situations during adjusting optimal value. It is case of linear dependence where process value will be insufficient during too low values or value will be too high and oscillation around required value will be too high and therefore controlling system will become unstable. It implies to conditions of successful adjusting of proportional component [1], [2], [8].

2. Integrating Component

The meaning of integrating part during regulation bases in reduction of permanent control deviation of P-controller. Integrating component determines process value according to time when control error does exist, see following equation:

$$u(t) = u_o + \frac{1}{T_1}\int_0^1 e(\tau)\, d\tau \tag{4}$$

We eliminate more or less permanent error of P-controller by integrating component, but oscillation will even emphasize during enlarging of integrating component. [1], [9].

3. Rate Response

The issue of oscillating in range of required value can be reduced by using rate response into regulated system. Rate response changes process value based on the change of rate response speed, see following equation:

$$u(t) = T_D \frac{de(t)}{dt} \tag{5}$$

Rate response attempts to compensate future errors based on preceding change of controlled variable. Principle lies in having impact against PI components in order, for instance, to prevent from exceeding setpoint within regulation. The exceeding would result in fatal consequences in application, so it uses just response rate which prevents from the exceeding, see Fig. 4. [1], [9], [10].

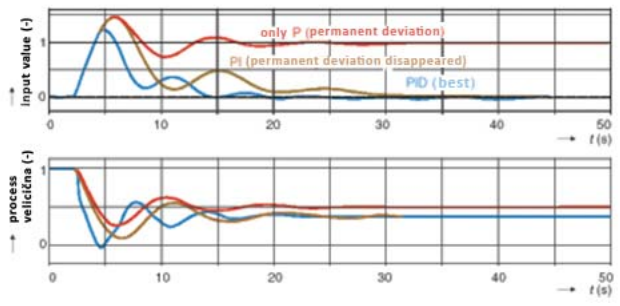


Fig. 4 Amplitude and phase frequency characteristics of controller

VI. Adjusting PID Regulation by Ziegler-Nichels Method

A. Open Controlling Circuit Method

We set parameters of PID controller by this procedure:

1. First Step

We record directly on controlled process or by means of simulation on computer a transitional response on transition input (0 až 100 %).

2. Second Step

Step response in shape „S“ is possible to characterize as parameters which we determine from transfer response. It is case of transport delay L and time constant (rise time) T. Situation is obvious from Fig. 1.

3. Third Step

We calculate parameters PI or PID controller according to Table I. [2], [4], [9]. Amplifying the process:

$$K_P = dy/du$$
$$K = T/(K_P \,.\, L) \tag{6}$$

TABLE I
Ziegler-Nicholas Adjusting, Open Controlling Circuit Method

	Proportional gain	Integral time constante	Derive time constante
PI controller	0,9 . K	3,3 . L	
PID controller	1,2 . K	2 . L	0,5 . L

B. Closed Controlling Circuit Method

We determine parameters PID controller by this procedure, see Fig. 5:

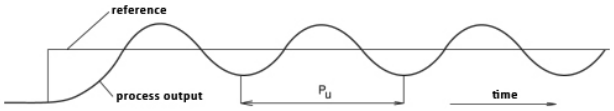


Fig. 5 Transitional characteristics for closed controlling circuit method

1. First Step

We completely delete integrating and derivational part of PID controller.

2. Second Step

We proceed to jump in setpoint and observe controlling response. We repeat jump of setpoint with increased or decreased proportional amplification until we record controlling response on stability limit. Controlling response is oscillating and oscillation amplitude does neither increase, nor decreases. It resembles to characteristics of stable oscillator. If we do not reach such state by amplifying of proportional amplification, then method cannot be used. The alternative for this step is to increase proportional amplification of controller step by step from zero and we observe when controlling response comes to stability limit. We have not changed setpoint value.

3. Third Step

We record value of proportional amplification K_u for stability limit and critical period P_u at these oscillations. We calculate parameters PI or PID controller according to Table II and we record it [2], [8], [9].

TABLE II
ZIEGLER-NICHOLAS ADJUSTING, CLOSED CONTROLLING CIRCUIT METHOD

	Proportional gain	Integral time constante	Derive time constante
PI controller	0,45 . Ku	Pu / 1,2	
PID controller	0,8 . Ku	Pu / 2	Pu / 8

VII. APPLICATION

We chose robot with wheel drive following black line on white surface as demonstration of infinite number of applications capable of using with the kit. Control drive of the whole robot supports two electric engines which belong to basic constructional kit. Programming went in freely available development environment and by programming language C#, too.

A. Line Detection Method

Algorithm implemented for move robot on dividing line of black and white line uses information collected from light sensors. Sensors detect range 0-100 (0 for black and 100 for white). Algorithm, as mentioned above, loads information from sensors and compares with setpoint which was adjusted on value 45 after sensors calibrating. Discovered difference between them is converted on process value which changes current speed of the engines driving wheels.

During testing of robotic platforms were created two types of controlling algorithms. Both algorithms use previously mentioned algorithm for calculation of actuating variable, but it varies the calculation of controlling deviation of values collected from sensors.

In both options there are sensors placed in such way that left sensor detects left side and right sensor detects right side of dividing line of black-and-white surface.

B. Algorithm no. 1

Calculation of process value in this solution corresponds with difference of information collected from sensors following black line. Realization of algorithm in development environment LEGO Mindstorms Home Edition is displayed in Fig. 6:

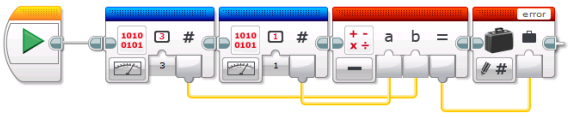


Fig. 6 Main part of "source code" for error calculation in Algorithm no.1

Demonstration of controlling code which was programmed using development environment MonoDevelop/Xamarin Studio C# follows there:

…

```
var ev3 = new Brick<Sensor,Sensor>("usb");
ev3.Connection.Open();
ev3.Sensor1 = new ColorSensor(ColorMode.Raw);
ev3.Sensor3 = new ColorSensor(ColorMode.Raw);
error =  ev3.Sensor1.read()-ev3.Sensor3.read();
```

…

C. Algorithm no. 2

The solution of controlling this algorithm is approached in different way than in previous case. Robot evaluates based on its position, which sensor appear in working state and based on the information regulation will control itself according to the sensor. In case that active sensor gets out of the position then the whole system starts to control itself using data collected by second sensor. Working state is defined by range in which current value is being collected by sensor. Realization of algorithm for calculation controlling deviation in development environment LEGO Mindstorms Home Edition is in Fig. 7:

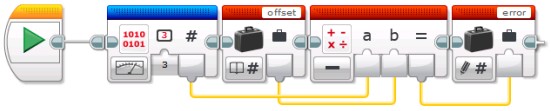


Fig. 7 Main part of "source code" for error calculation in Algorithm no.2

Demonstration of controlling code in programming language C# is that:

…

```
var ev3 = new Brick<Sensor,Sensor>("usb");
ev3.Connection.Open();
ev3.Sensor3 = new ColorSensor(ColorMode.Raw);
error =  ev3.Sensor3.read()-offset
```

…

VIII. CONCLUSION

In this paper we have described the possibility of simulation of robotics systems with Lego Mindstorms kit. As the main benefit of the solution is possibility to simulate, programme and test robotics systems. We see as motivation for our approach is to enable to simulate robotics systems in low-cost way compared to professional kits for simulation robotics systems. We proved that using of Lego Mindstorms kit leads to improve efficiency and cut costs. We have identified in detail two possible solutions of adjusting of PID controller components by two ways. Using by empirical methods is applied to estimate parameters of PID controller. Firstly we dealt with trial-and-error method to adjust PID controller components, secondly we used alternative way of adjusting by Ziegler-Nichols method. Discovering of correct combination of parameters may seem to be difficult, however it was spent a lot of time to develop code and repetitive modifications the rank of controller components. We are convinced that Lego Mindstorms kit will be considered for quality and low-cost tool at the same time to simulate robotics systems in order to create industrial applications

ACKNOWLEDGMENT

This work was supported by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2014/017 and No. IGA/FAI/2014/007.

REFERENCES

[1] Švarc, Ivan. Automatické řízení: First International Conference, SENSAPPEAL 2009, Athens, Greece, September 25, 2009, revised selected papers. Vyd. 2. Brno: Akademické nakladatelství CERM, 2011, vi, 348 p. ISBN 978-80-214-4398-3.
[2] Švarc, Ivan. Automatizace: automatické řízení. 2. dopln. vyd. Brno: CERM, 2005, 262 p. ISBN 80-214-2943-7.
[3] O´Dwyer, Aidan. Handbook of PI and PID Controller Tuning Rules. 2009. ISBN 978-1848162426.
[4] Visioli, Antonio. Practical PID control. London: Springer, c2006, xviii, 310 p. ISBN 18-462-8586-0.
[5] Najim, Kaddour. Control of continuous linear systems. London: ISTE, 2006. ISBN 978-047-0612-347
[6] Lego mindstorms ev3 discovery book: a beginner's guide to building and programming robots. S.l.: O'Reilly Media, 2014. ISBN 978-159-3275-327.
[7] Rollins, Mark. Beginning Lego Mindstorm EV3. New York: Apress, 2014. ISBN 1430264365
[8] Wescott, Tim. Applied control theory for embedded systems. Burlington, MA: Newnes, c2006, ix, 303 p. ISBN 978-075-0678-391.
[9] Dorf, Richard C a Robert H Bishop. Modern control systems. 12th ed. Upper Saddle River: Prentice Hall, c2011, xxii, 1082 p. ISBN 978-0-13-602458-3.
[10] Ogata, Katsuhiko. Modern control engineering. 5th ed. Boston: Prentice Hall, c2010, x, 894 p. ISBN 978-0-13-615673-4.
[11] MonoBrick.DK [online]. 2014 [cit. 2014-05-06]. Dostupné z: http://www.monobrick.dk/

Miroslav Popelka studied at the Tomas Bata University in Zlín, Czech Republic, where he obtained his master degree in Computer and Communication Systems in 2013. He now attends PhD. study in the Department of Automation and Control Engineering of the Tomas Bata University in Zlín. His research interests focus on ultrasonic signal processing. He is currently working on programming ultrasound imaging for mobile robot systems.

Jakub Nožička studied at the Tomas Bata University in Zlín, Czech Republic, where he obtained his master degree in Computer and Communication Systems in 2013. He now attends PhD. study in the Department of Informatics and Artificial Intelligence of the Tomas Bata University in Zlín. His research interests focus on methods of detection intruders in wireless networks. He is currently working on programming web aplications.