Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 7, No. 4, December 2019, pp.1868-1878

Dynamic analysis, modeling and control of the LEGO EV3 modular mobile platform

Faruk Bevrnja[1], Mustafa Bevrnja[1], Marin Petrović[1]
[1] Department of Mechanics, Mechanical Engineering Faculty, University of Sarajevo

ABSTRACT

Due to the increased popularity of robotic systems and their more frequent application, some companies have opted to incorporate modular robotic systems in their assortment. The advantages of such systems are great flexibility in terms of combining components, relatively easy programmability, a wide range of functions that can be performed, the availability of components, as well as modularity in terms of functional and structural connectivity. The disadvantages are reflected by the fact that these systems are not optimized for a particular application. The precision and accuracy of such systems are substantially less than those of the system created exclusively for a particular application. Gaps and dead strokes of moving parts are very influential on the performance of functions and there is relatively little autonomy to such systems. The problem of mutual inarticulation of motors was introduced, and a solution was given to overcome these problems. The observations concerning the aforementioned problems have been explained and the most important features of this approach to robotic systems highlighted.

Keywords: EV3, LEGO, Simulink, Modeling, Modular systems

Corresponding Author:

Marin Petrovic,
Mechanical Engineering Faculty,
University of Sarajevo,
Vilsonovo setaliste 9, 71000 Sarajevo, Bosnia and Herzegovina.
E-mail: petrovic@mef.unsa.ba

1. Introduction

Modular platforms are increasingly used for research and educational purposes. There are numerous examples of such systems, such as: Arduino, Adimec, LEGO, NAO, Raspberry, etc. The aforementioned manufacturers strive to provide reliable modular solutions that, by combining them, can achieve a virtually infinite set of functional systems. The advantages of this approach are that individual components are easily accessible, mutually compatible, relatively easily programmable, robust and tailored so that, in addition to being executives, they also play a structural role in the design of the desired system. Some of the above characteristics can be considered as advantages, but also disadvantages. The proof of this assertion is that modular platforms are inherently unadapted to individual solutions and that systems made of them are not optimized from a constructive or even functional point of view. Namely, for a modular platform to have as wide application as possible, which is actually its primary characteristic, individual components must be designed to be as general in character as possible in terms of the range of functions they perform and how they are combined with other components. Some of the disadvantages are steady-state error, drift, nonlinearities in some parts of the functional range, insufficient sensitivity, too small step and large sampling interval.
Several modular platforms are available and outlined in literature. In a new motion planning algorithm [1] users can interact with the environment and the robot in real-time via a web camera, but without tests of reliability. The conceptual model-based approach for modular complex system development can be regarded

as the main contribution of Suri et al. [2], which can be applied to industrial cyber-physical systems. These kinds of platforms are good for education, in specific tasks like teaching many theoretical algorithms [3-6]. Several authors have dedicated their work to finding optimal methods for controlling mobile robotic devices [7-10]. To be able to do these tasks properly, equipment must be reliable and accurate. This paper is focused on calibration and control of specific components. The task is performed by analyzing the kinematics and dynamics of the LEGO platform. A set of equations is formed for the general case of plane motion in a plane, concrete platform trajectory programmed and current flow measured through servo / dc motors on LEGO platform. The scalar function of the motor moments is predicted required for the platform to follow the given trajectories (and laws of motion) and compared with the measured values from the previous point. Matlab / Simulink Realtime Workshop software is used, exploring real-time data transfer capabilities to a remote computer, using the aforementioned software. The possibilities of forming a controller (PID, inverse model, fuzzy) in Simulink are examined, managing the platform from a remote computer. The idea of this paper is to perform a research whether it is possible to make an autonomous robot using easily accessible commercially available hardware and which variables have to be taken into account when creating it.
The specific LEGO EV3 platform studied consists of the following components:

- EV3 Cube - The main processor unit, whose features are 300 MHz MHz processor, 64 Mb RAM, support for Bluetooth and WiFi technology, USB 2.0 communication and micro SD card support. The integrated peripheral components are screen, speakers, state lights and control buttons;
- Large servo motor (actuator) - maximum operating speed of 170 rpm, torque of 0.2 Nm and stopping torque of 0.4 Nm. It is positioned in the engine case an integrated encoder, a rotation meter, whose step is 1 degree of rotation and least sampling time 0.001 s;
- Medium-size servo motor - maximum speed of 250 rpm, running torque 0.08 Nm and stopping torque of 0.12 Nm, also with integrated encoder, identical to that of a large engine;
- Color sensor - supports three operating modes (with factory firmware), which are ambient light recording, registration of the reflected light intensity, since the sensor has the multicolor LED used as a light source, color recognition from the programmed color palette (eight colors);
- Sonar - ultrasonic sensor;
- Single-axis gyro sensor;
- Touch sensor - button with three modes, generating an output non-zero signal such as:
  - the signal is greater than zero as the force acts on the sensor,
  - the signal is greater than zero as long as the force does not act on the sensor, it can be concluded that the previous two modes give a pulse signal if the effect is on the sensor pulse, that is, the sensor acts as a button,
  - changing the initial state gives the signal greater than zero - the sensor behaves as a switch.

2. Engine testing

In order to measure the electric current and voltage, test cases were made to obtain the most accurate results and the desired motor characteristics. Test cases, depending on the purpose, meant defining the input voltage, current, load, and engine speed.

2.1. Test case no. 1 - Characteristics of unloaded operation

Testing conditions are as follows: the motor is powered by a variable, regulated source. The ammeter measures the current flowing through the motor and the voltmeter monitors the voltage throughout the range. At the maximum rated speed of a large motor - 175 rpm, a current of 60 mA was measured, at a voltage of 9 V. Furthermore, this test was extended to obtain a dependence of the speed of rotation and voltage. Given that it is a DC motor, and provided it is unloaded or subjected to a relatively low load of constant intensity, the speed of rotation is proportional to the voltage to which it is subjected, as can be seen in Fig. 1. Also, it is important to note that the current under load is very little dependent on the voltage (ideally, the load is zero through the motor windings).

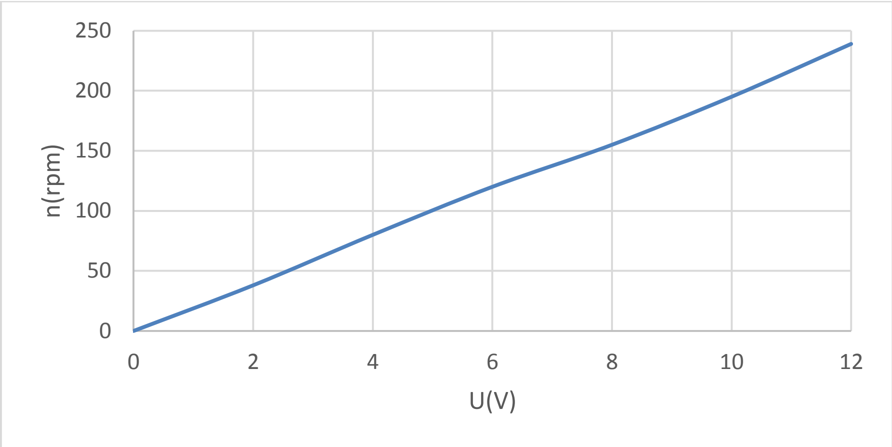

Figure 1. Voltage vs. engine speed in rpm

2.2. Test case no. 2 – Blocked motor characteristics

The current was measured while the motor output shaft was jammed manually. The stopping moment is determined from the maximum mass that can be raised under the given conditions. Measured current value for the case when the stopping moment is 0.43 Nm amounted to 1.8 A, at a voltage of 9 V.

2.3. Test case no. 3 - Characteristics when loaded

The mechanical power supplied by the engine can be calculated from the time it takes for some, predefined load, to be lifted by a certain height. The drum with string windings is directly connected to the motor shaft and the load is hung at the end of the string. Using two electromechanical relays, the NXT cube controlled the motors in three cases: lifting, level retention, braking. The results obtained are given in Table 1 and presented in Fig. 2.

Table 1. The speed of rotation and the current intensity

Voltage	Torque	Revolution	Current	Mechanical power	Electric power	Efficiency
4.5 V	17.3 Ncm	24 min^{-1}	0.69 A	0.43 W	3.10 W	14 %
6.0 V	17.3 Ncm	51 min^{-1}	0.69 A	0.92 W	4.14 W	22 %
7.5 V	17.3 Ncm	78 min^{-1}	0.69 A	1.41 W	5.17 W	27 %
9.0 V	17.3 Ncm	105 min^{-1}	0.69 A	1.90 W	6.21 W	31 %
10.5 V	17.3 Ncm	132 min^{-1}	0.69 A	2.39 W	7.24 W	33 %
12.0 V	17.3 Ncm	153 min^{-1}	0.69 A	2.77 W	8.28 W	33 %

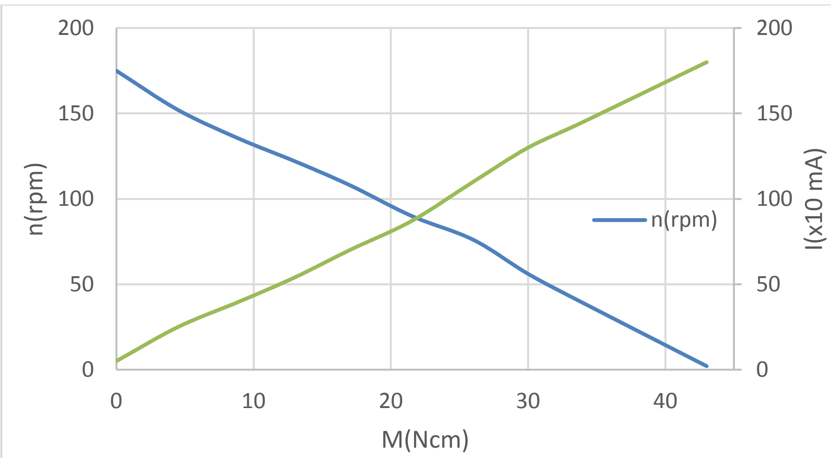

Figure 2. Power-to-torque ratio and torque-to-torque ratio for EV3 engine

3. Platform movement in plane

Platform movement in a plane is controlled by a software that defines the required speeds of individual wheels in time. Although the software manages the angular velocity of the wheels, knowing the translational velocity of the wheel centers is crucial for analyzing the motion of the platform in the plane. Given that the platform moves at low speed and has wheels with rubber rims, it is accepted that longitudinal wheel slippage and lateral slip when cornering can be ignored. This makes it possible to state that the speed of movement of the center of the wheel perpendicular to the axis of its rotation and angular velocity of a wheel are proportional to the translational velocity of its center:

$$v_R = \omega_R R\,; \qquad v_L = \omega_L R \tag{1}$$

where v_R is right wheel line speed (m/s), v_L is left wheel line speed (m/s), ω_R is angular velocity of right wheel (s^{-1}), ω_L is angular velocity of left wheel (s^{-1}) and R is wheel radius (m).

Therefore, velocity of platform's center is given by equation (2), whereas angular velocity of platform can be calculated using equation (3):

$$v_0 = (v_R + v_L)/2 \tag{2}$$

$$\omega = (v_R - v_L)/w \tag{3}$$

where v_0 is velocity of platform's center (m/s), ω is angular velocity of the platform (s^{-1}) and w is wheel track width (m).

If the desired velocity of the center of the platform v and the desired angular velocity of the platform ω are known, the necessary velocity points v_R and v_L can be determined using (4). Therefore, the required angular velocities of the wheels are given by (5).

$$v_R = v + \omega w/2\,; \quad v_L = v - \omega w/2 \tag{4}$$

$$\omega_R = (2v + \omega w)/(2R)\,; \quad \omega_L = (2v - \omega w)/(2R) \tag{5}$$

Based on the theory of plane motion and previously written equations, the functions of the velocity of the platform depending on the position on the trajectory and current radius of the curve can be determined. This allows one to define a lower speed for sharp turns and curves of small radius, and a higher one for more straight sections of the trajectory, which is a great advantage.

This implies that the trajectory along which the platform is supposed to move then goes beyond x and y coordinates, depend on some parameter p:

$$x = x(p) \text{ i } y = y(p) \tag{6}$$

Moreover, the expression for the radius of curvature R_k of the trajectory should be derived, as follows:

$$ds = R_k d\varphi \tag{7}$$

Now the following may be written:

$$ds^2 = dx^2 + dy^2 \;\rightarrow\; ds = \sqrt{dx^2 + dy^2} \;\rightarrow\; ds = \sqrt{dx^2 + dy^2}\,\frac{dp}{dp} \;\rightarrow\; ds = \sqrt{\left(\frac{dx}{dp}\right)^2 + \left(\frac{dy}{dp}\right)^2}\,dp \tag{8}$$

Equation (8) can now be represented as:

$$ds = \sqrt{\dot{x}^2 + \dot{y}^2}\,dp \tag{9}$$

Furthermore, the dependencies of velocity from control parameter p (where $v = v(p)$) need not be invented for each curve that the platform needs to cross, but rather defines a rule by which the platform adjusts its speed to the geometry of the trajectory. The following is one possible establishment of a connection $v = v(R_k)$, where R_k is a radius of curvature. That could describe the real behavior of the platform. The requirements adopted for this example are that the velocity of the platform when moving in a direction is $v_{\max}$, and when the trajectory is broken, the platform stops completely. In other words, when $R_k = 0$, it is necessary that $v = 0$, and when $R_k = \infty$, $v = v_{\max}$. With some knowledge of the functions of a variable, the following family of relations can be obtained:

$$v = v_{\max} R_k / (R_k + c) \tag{10}$$

where c is a parameter that determines how fast the speed increases with increasing the radius of curvature. After a brief analysis, it can be seen that the parameter c equals the radius of the curve at which the velocity of motion will be equal to half the maximum velocity. In this example, it is adopted that the parameter c is as large as the wheel track w.

4. Model

The modeling and programming of the system was performed in the Simulink environment, which is a Matlab graphical object representation, in addition to previously performed work using the free-software platform [11].

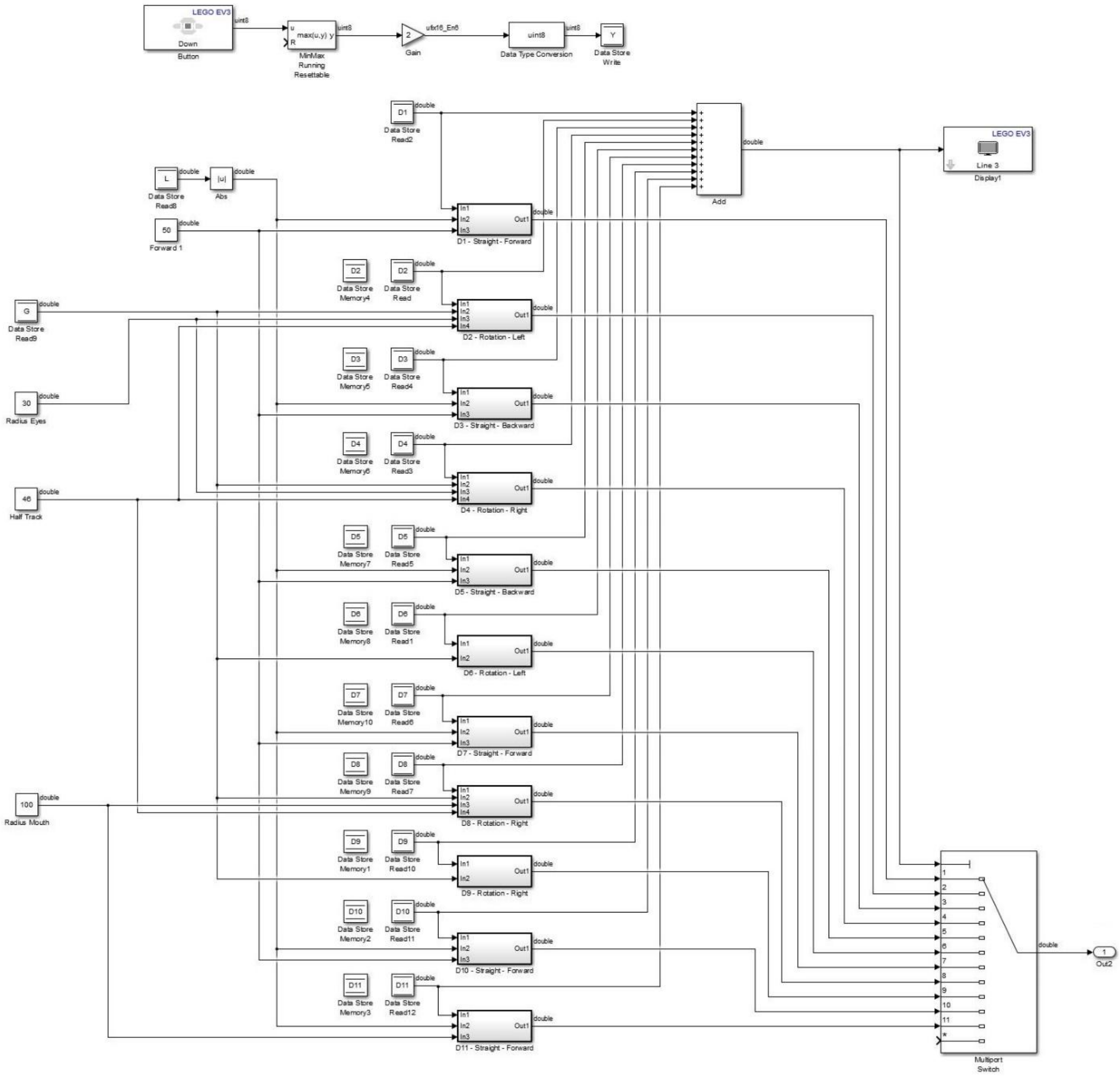

Figure 3. Subsystem for drawing

Simulink has the ability to calculate and execute a very large number of functions at the same time. Therefore, given the above fact, it was necessary to sequence the execution of those functions. The program is divided into two main parts: orientation (serves to orient the system in space based on a predefined marker) and drawing (performs the main function of the program, that is, draws / writes based on the defined input). The program contains eight subsystems: case switcher subsystem, orientation subsystem, drawing subsystem, sensor subsystem, length measurement subsystem, encoder subsystem, gyro sensor subsystem and color sensor subsystem. An example of one of the subsystems is shown in Fig. 3.

5. Real-time data transmission

Matlab's hardware support package for LEGO EV3, in addition to supporting modeling and programming with all the blocks supported by the EV3 cube, has the ability to link Simulink to the EV3 cube while executing the program. The offered ways of executing the program are:

1. normal - the program can be independently run on a computer to test its correctness and can be sent to the EV3 cube to test the functionality of the program and
2. external - the program is executed on the computer, and by running the program on the computer, two-way communication with the EV3 cube is also achieved. This allows the computer to manage operations and the cube sends the current motor and sensor parameters to the computer.

Input signal to the motors and the measured length at each stage of movement are shown in Fig. 4 and Fig. 5. These signals are recorded in real time.

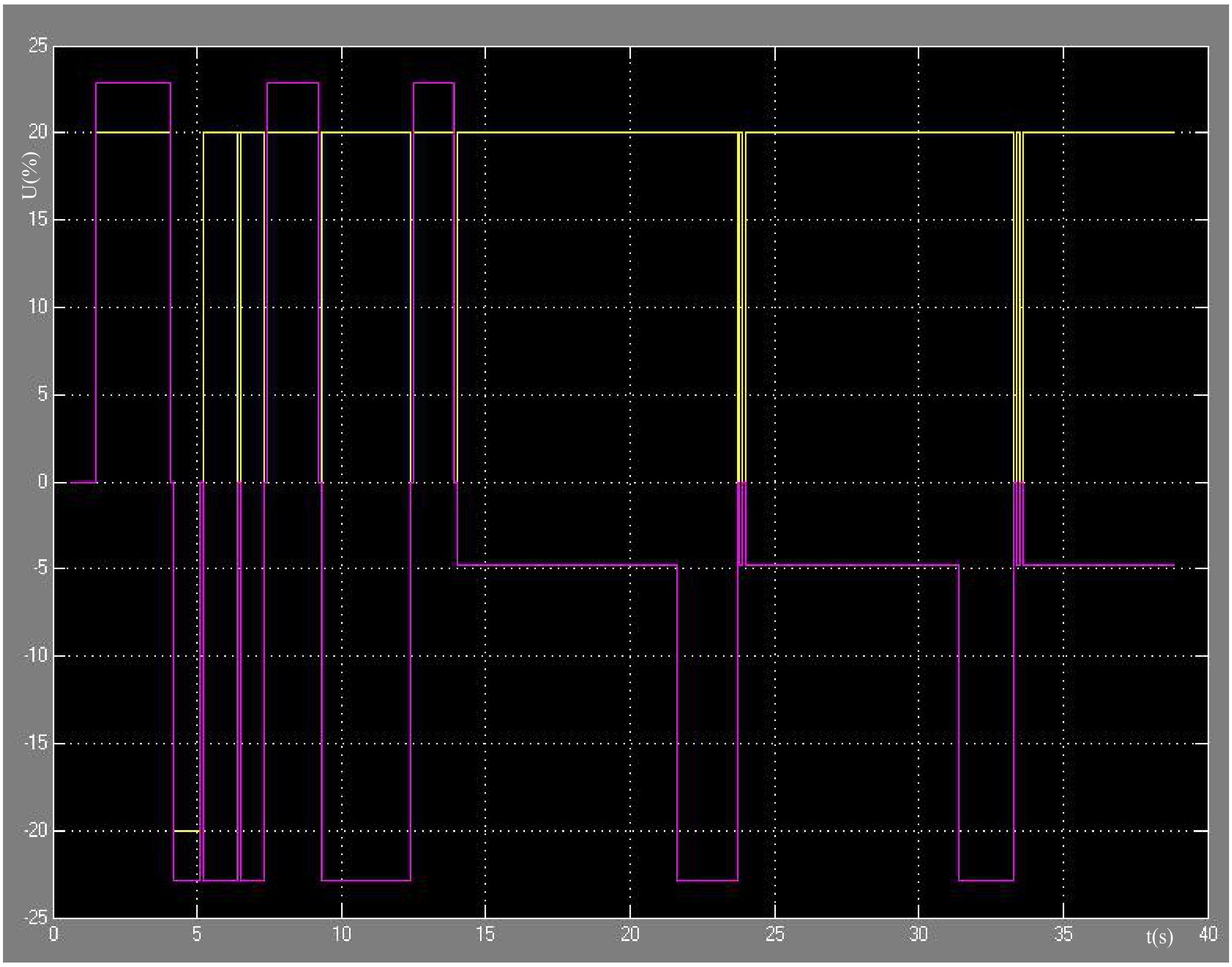

Figure 4. Input signals to blocks for motors

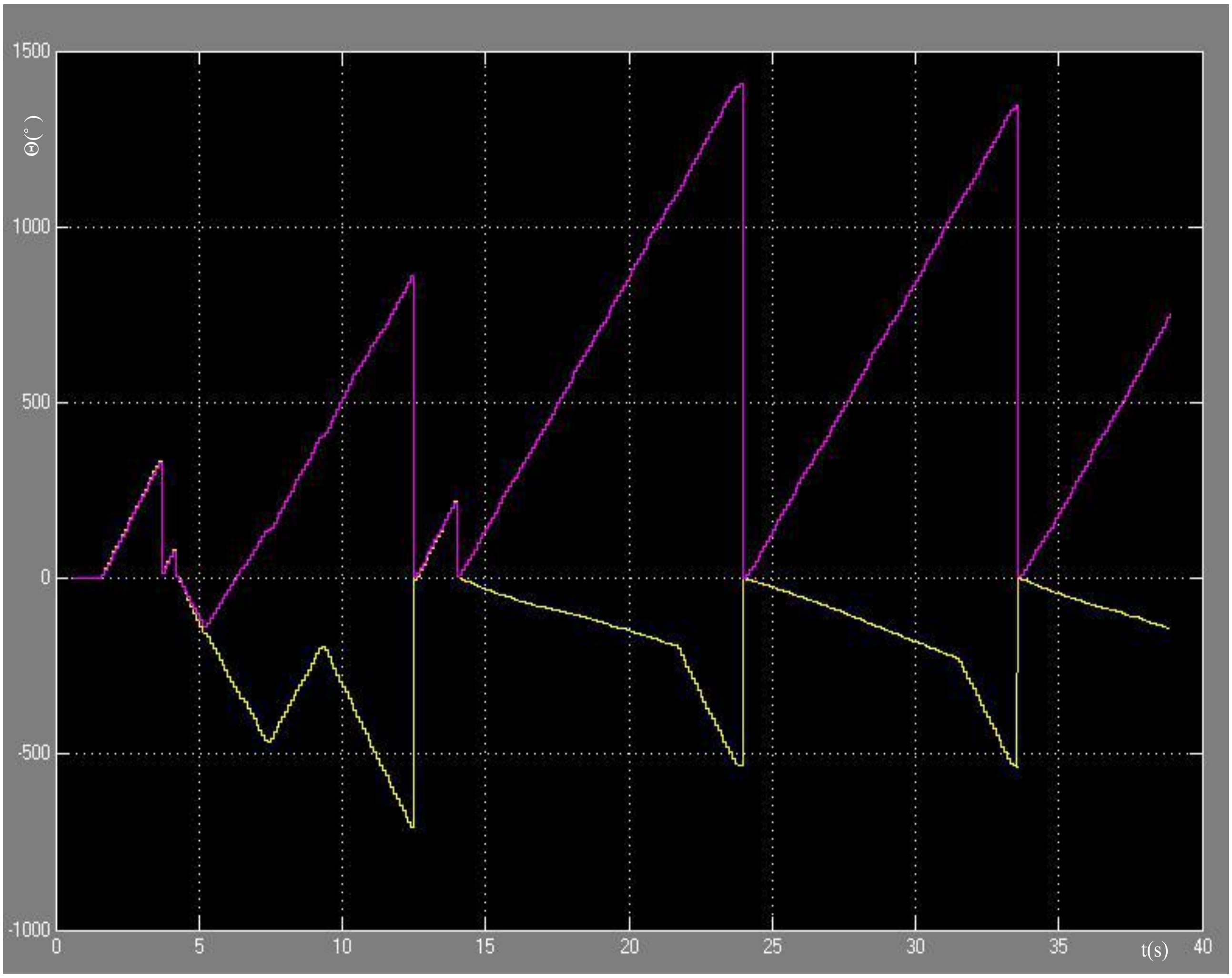

Figure 5. Output summed signals from encoder

The reason different maximum values of left and right motor speeds are registered lies in the fact that the motors are not identical and therefore one of the input signals is corrected. Namely, for the same nominal input, the motors returned different real speeds of motion, thus causing movement problem. This problem is solved by signaling on one motor based on the reading of the rotation angle from the individual wheels. Both engines, as mentioned previously, have built in encoders. Both encoders give accurate readings on the basis of which a control signal for the controller is formed. Apparently, due to the relatively small angular acceleration of the wheels, the slip can be neglected. Fig. 5 shows that the readings are almost identical on the sections of the graphs where the motion of the motor is of the same direction and intensity. This represents the proof the regulator has been successfully optimized. The sections on which the signals are on opposite sides are abscissa readings of the stages in which the system is rotated by a curve whose current radius is less than half the track of the wheels.

6. Results - calibration and controller

The design of the controller was preceded by a test of the output speeds of the engines used. For this purpose, a program was created giving the motors an equal input signal and measuring the output speeds. The layout of the program is given in Fig. 6. In the first phase, the program shown above gives the engines the same input signal and reads the differences in speeds. The input signal is incremented by 10% step by step and thus measures the speed differences for inputs from 0 to 100% through ten steps. The system is modeled to increase the input signal, hold this value for one second, and then start measuring to stabilize transient processes. Due to the low mass of the parts, a delay time of one second is quite sufficient. After the defined reading period expires, the measurement stops and the signal increases again. This gradual reading gives an overview of the difference in speeds between the engines used.

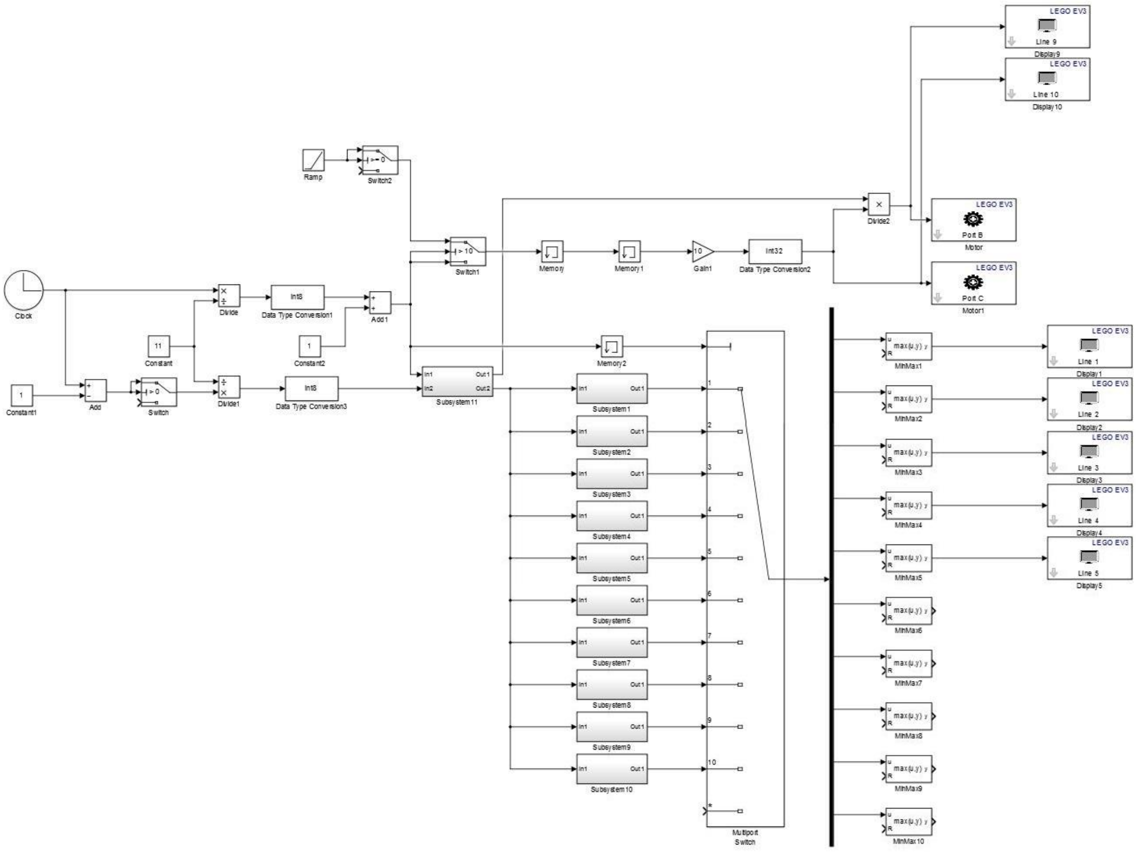

Figure 6. Layout of the program used to calibrate and design the controller

Fig. 7 shows a graphical representation of the difference between these speeds. The discrepancies are obvious, and are caused by many factors. Some of these factors are the warming of the engine, the type of power supply, i.e. whether the test was performed by an external constant power source or by the use of batteries, etc. It is noticeable that the character of the lines is very similar, indicating the persistence of the difference under different external conditions.

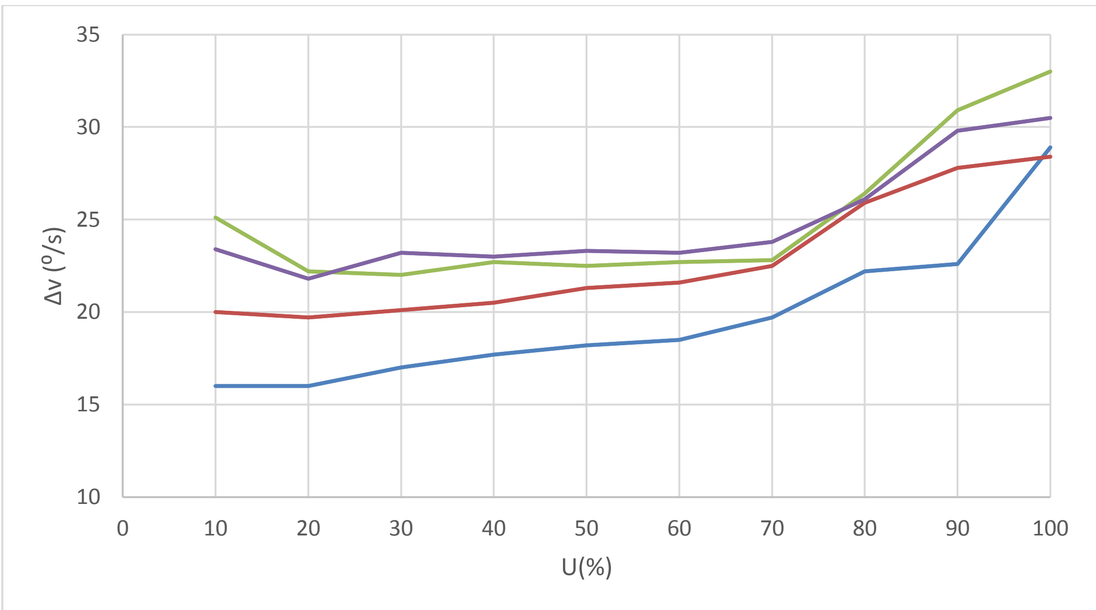

Figure 7. Difference of engine output speeds in several consecutive measurements

After establishing the existence of the difference and its character, it can be concluded that the speed correction of one of the two motors could not be performed with constant amplification and that it was necessary to form a regulator. Fig. 8 shows the layout of the subsystem, the previously shown controller-related program.

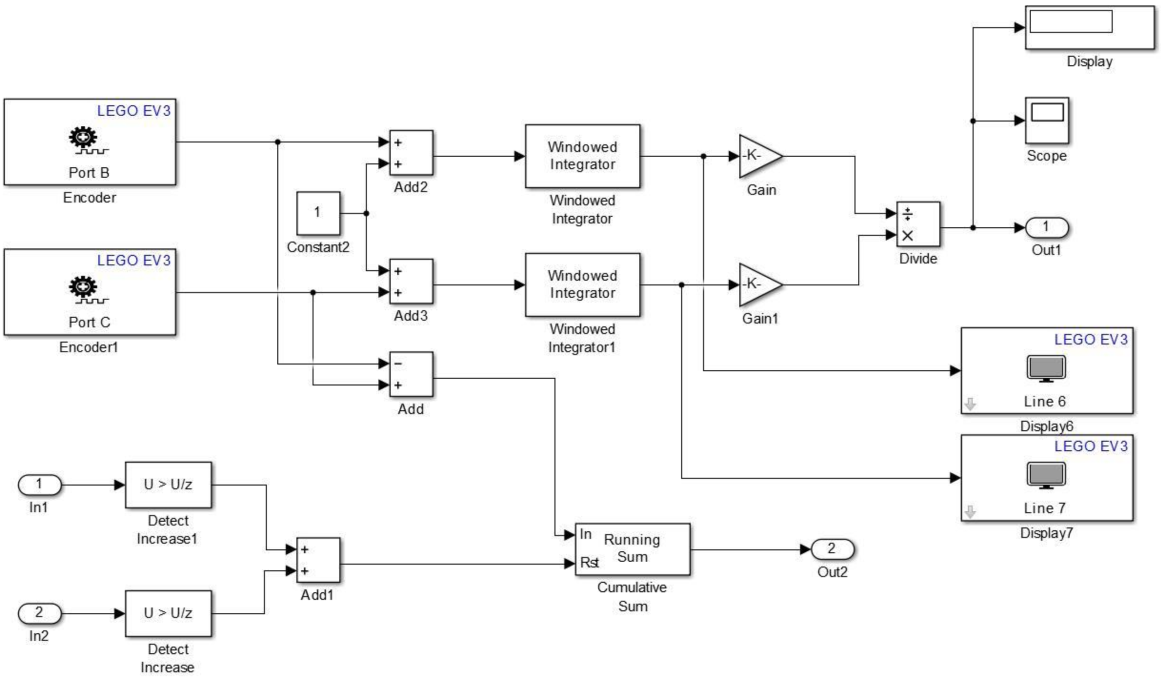

Figure 8. Control subsystem

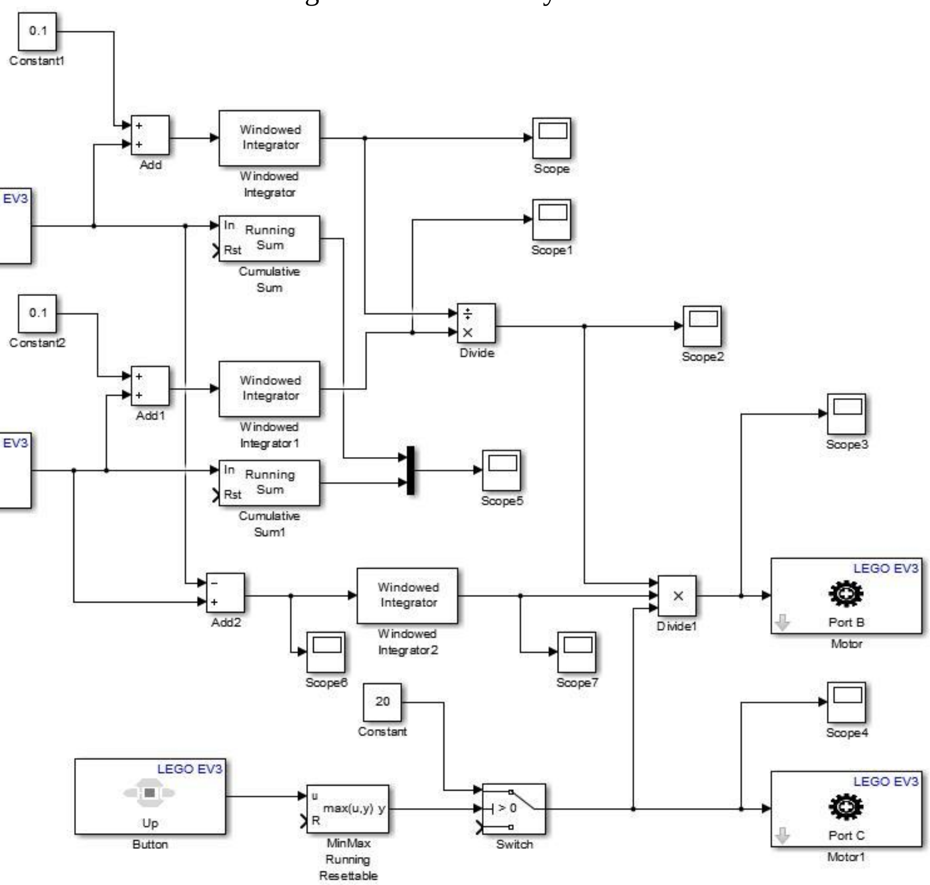

Figure 9. Attempt to run an integrated signal controller

The input signal of one of the motors is multiplied by the coefficient of the speed quotient of the two motors. This aforementioned quotient is obtained by dividing a signal that is integrated at a defined displacement interval after being read from the encoder. This method led to a significantly reduced speed difference. The system was not compromised and the response time was acceptable for a given application. Moreover, an attempt was made to build a controller that, in addition to the aforementioned parameters, would use an integrated instantaneous fault signal. However, as the signal integration leads to an increase in the instability of the system, one wheel has a good, constant speed and the other variable and highly oscillatory character. The layout of the program is given in Fig. 9. Due to the above facts, the first way of controller modeling was chosen.

7. Conclusions

LEGO EV3 platform has been chosen as a test platform for testing the potential of modular robotic systems and some of its features have been tested. Some of the possibilities tested were accomplished using the Matlab Simulink environment. It has been proven that such systems may require realistic robotic conditions to evaluate any robotic platform such as programmability, ability to combine with other systems and components, capability communications with some commercial software, real-time data transfer capability and, most importantly, consistent execution of the desired task. It has been shown that it is necessary to make numerous adjustments within a single model in order for all components to communicate seamlessly. These adjustments include the conversion of the types of data used, sequencing, multiplexing, setting control conditions, signals and sizes.

The biggest problem was to harmonize the different phases of the program, since they needed to be executed in precisely defined order. The way to overcome these problems is elaborated in detail and reasoned. One of the goals was to examine the possibility of establishing real-time data transmission, which was successfully achieved. However, the capabilities allowed even the budget to be executed on the mainframe, and only the executable commands were executed through the EV3 cube. A very significant problem posed the physical differences between the two seemingly identical engines. For the same input signal, the motors gave up to 10% different output, which is unacceptable for certain precision actions such as writing or drawing. Formation testing was necessary and required some kind of regulator. The formation of the controller concerned the determination of the most favorable size which would be monitored and compared to another relevant magnitude and therefore the output signals of the encoder blocks were observed. Despite the many limiting factors that modular platforms face, they have very large range of applications. The most obvious disadvantages are insufficient precision of some components, high deadweight of moving parts, inability to perform repeatedly certain functions that depend on the surrounding conditions and cannot be influenced by the given components and the relatively high cost of individual components. Notwithstanding the wide range of capabilities of some components, which is primarily related to the EV3 cube, the capabilities of the cube are limited by a small number of different functional components, which again have a very narrow application and the durability of plastic and metal parts.

For the following phase of this research, it is necessary to introduce new hardware components. Software solutions alone are insufficient. The greatest challenge in the current setup is the control of power, i.e. the voltage of the current supplied to the motors, as well as the inability to predict the variability of output as a function of the temperature of the internal components. By adding temperature sensors, additional current regulators, the ability to monitor impact variables across the entire device would be improved. The results obtained would allow the creation of fully autonomous devices with complete internal controls without the need for connection to external devices.

References

[1] N. Montés, N. Rosillo, M.C. Mora and L. Hilario, “Real-Time Matlab-Simulink-Lego EV3 Framework for Teaching Robotics Subjects”, in: W. Lepuschitz, M. Merdan, G. Koppensteiner, R. Balogh, D. Obdržálek (eds) Robotics in Education, *Advances in Intelligent Systems and Computing*, vol. 829, pp. 230-240. Cham: Springer Nature Switzerland AG, 2019.

[2] K. Suri, A. Cuccuru, J. Cadavid, S. Gerard, W. Gaaloul and S. Tata, “Model-based Development of Modular Complex Systems for Accomplishing System Integration for Industry 4.0”, In *Proceedings of the*

5th International Conference on Model-Driven Engineering and Software Development MODELSWARD 2017, Porto, 2017.

[3] F. Klassner, J.C. Peyton-Jones and K. Lehmer, "Genetic Algorithms with Lego Mindstorms and Matlab", In *Proceedings of the 25th The Florida Artificial Intelligence Research Society* FLAIRS 2012, Florida, 2012.

[4] J. Ding, Z. Li and T. Pan, "Control System Teaching and Experiment Using LEGO MINDSTORMS NXT Robot", *International Journal of Information and Education Technology*, vol. 7, no. 4, pp. 309-313, 2017.

[5] C. Lapusan, O. Hancu, C. Rad, L. Dache and V. Maties, "Integrated learning platform based on Lego NXT and Matlab for teaching mechatronics", In *8th International Conference on Electronics, Computers and Artificial Intelligence* ECAI 2016, Ploiesti, 2016.

[6] N. Özüorçun and H. Bicen, "Does the Inclusion of Robots Affect Engineering Students, Achievement in Computer Programming Courses", *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 13, no. 8, pp. 4779-4787, 2017.

[7] R. Singh and T.K. Bera, "Obstacle avoidance of a humanoid mobile robot based on multi-sensor information", *World Journal of Modelling and Simulation*, vol. 15, no. 2, pp. 116-127, 2019.

[8] R. Kaur and A. Singh, "Fuzzy logic: an overview of different application areas", *Advances and Applications in Mathematical Sciences*, vol. 18, no. 8, pp. 677-689, 2019.

[9] A.S. Fadeev, A.Y. Zarnitsyn, A.V. Tsavnin and A.S. Belyaev, "Cyber-physical system prototype development for control of mobile robots group for general mission accomplishment", *International Scientific and Practical Conference Modeling in Education 2019 Proceedings*, vol. 2195, no. 1, 2019.

[10] F. Valdez, O. Castillo, C. Caraveo and C. Peraza, "Comparative Study of the Conventional Mathematical and Fuzzy Logic Controllers for Velocity Regulation", *Axioms*, vol. 8, no. 2, pp. 53-71, 2019.

[11] A.M. González-Vargas, J.M. Serna-Ramirez, C. Fory-Aguirre, A. Ojeda- Misses, J.M. Cardona- Ordoñez, J. Tombé- Andrade and A. Soria- López, "A low-cost, free-software platform with hard real-time performance for control engineering education", *Computer Applications in Engineering Education*, vol. 27, no. 2, pp. 406-418, 2019.