

Remote control of LEGO Mindstorms NXT motors programmed in Python

Nikola Jović, Vladimir Cvjetković and Milan Matijević, *University of Kragujevac*

Abstract — Technical solution for control of Lego Mindstorms NXT motors using NXT Python library for correction of some NXT Python flaws found in previous versions is presented in this paper. Educational potential of technical solution that is also used in NXT Web Laboratory is demonstrated. Application of approach described in this paper offers many benefits such as financial savings and greater educational potential comparing to using of software tools like LabVIEW and Matlab.

Key words — Web Laboratory, NXT Python, DC motor control, LEGO Mindstorms NXT, Engineering education.

I. INTRODUCTION

LEGO Mindstorms NXT is a LEGO company system global product, which is present on the market for eight years, among other purposes found its place as a mean of education for all educational levels. It is available at a retail price of 290 \$ in the USA, and a little more expensive in Serbia. Although relatively cheap, the LEGO NXT is not a standard mean of education at domestic lower educational levels for two main reasons: 1) Financing of means for educational support is always an open question, and, 2) There is no domestic developed methodology for such educational support, nor for pupils, nor for teachers, comparing to some other countries like Slovenia for instance. On university level, owing to various EU donation and credit programs of help for Serbia in the first place, the material base of education and research was significantly improved in the last ten years, so that the transfer of educational practice from the world universities was partly enabled. [1-6].

In [1] one aspect of LEGO Mindstorms NXT application at the University of Kragujevac is described, with possible similar applications for other institutions and levels of education. The base of typical Python software programs was developed for education of students, enabling relatively quick access for experimenting with LEGO platform, without requirements and support of commercial software tools. But using of NXT Python libraries [7] exhibits the unexpected flaw with control NXT LEGO DC motors: the motor static characteristics [8] becomes distorted with “dead zone”

Nikola Jović – Fakultet inženjerskih nauka Univerziteta u Kragujevcu, Sestre Janjić 6, 34000 Kragujevac, Srbija (e-mail: nikolajovic@live.com).

Vladimir Cvjetković – Prirodno-matematički fakultet u Kragujevcu, Univerzitet u Kragujevcu, Radoja Domanović 12, 34000 Kragujevac, Srbija (e-mail: vladimir@kg.ac.rs).

Milan Matijević – Fakultet inženjerskih nauka Univerziteta u Kragujevcu, Sestre Janjić 6, 34000 Kragujevac, Srbija (e-mail: matijevic@kg.ac.rs).

occupying $\pm 50\%$ of the available range, and which is not present when using the other software tools like NXC, LabVIEW, Matlab/Simulink. The main contribution of this paper is removing of the mentioned flaw [7], and creation of web portal for complete support of student laboratory exercises with LEGO Mindstorms NXT, including detailed eLearning materials for application of NXT Python, template Python files as the base of laboratory exercises, on line access and control of laboratory equipment consisting of LEGO NXT components as in [9]. The purpose of such an approach is to leverage the availability of laboratory resources in education with teaching subjects having high number of attendants.

Besides, the paper also discusses the mapping of teaching methodology at world famous engineering schools to our domestic conditions and nature of current subjects with already defined and accredited contents within our educational system. The illustrative scenario of theoretical concepts is based on basic LEGO Mindstorms NXT set with sensors and actuators (DC motors).

II. LEGO MINDSTORMS NXT AND LEGO NXT MOTORS

The basic LEGO Mindstorms NXT set consisting of programmable unit, two touch sensors, sensors for color / light intensity, ultra sound distance sensor, three DC motors with integrated encoder and redactor, is shown in Fig 1.



Fig. 1. LEGO Mindstorms NXT 8540: programmable unit, three DC motors with integrated encoders, two touch sensors, color/light sensor, and ultrasonic distance sensor.

Minicomputer or programmable unit (popularly called the “brick” [6]) consists of following significant components:

- 1) The main processor with working frequency of 48 MHz, 32 bit ARM microcontroller Atmel AT91SAM7S256, 256KB of flash memory of, 64KB of RAM memory and USB interface;
- 2) The auxiliary processor with working frequency of 8 MHz,

- 8 - bit Atmel AVR processor ATmega48, 4KB of flash memory and 512B of RAM memory;
- 3) 4 analog inputs for sensors;
- 4) 3 PWM outputs for driving of LEGO DC motors with channel for reading of encoders integrated in housing of LEGO DC motors;
- 5) Wireless Bluetooth CSR BlueCore™ 4 v2.0 +EDR system, which support serial input, 47KB of RAM memory, 8MB of flash memory, working frequency of 26MHz;
- 6) USB 2.0 for data exchange with maximal speed of 12 Mbit/s;
- 7) LCD monochromatic display with resolution of 100x64 pixels;
- 8) 2-16KHz loudspeaker with 8-bit resolution;
- 9) Power supply, consisting of 6 AA batteries (9V).

Three output connectors for motor control designated as, A, B and C, while the input connectors for sensors are designated as 1, 2, 3 and 4. Output and input connectors are standard 6 wire cables.

The “brick” outputs can be connected with actuators (Fig 2, [9]), with commercial name “LEGO Mindstorms NXT Interactive Servo Motor (9842)”. It is an integrated product consisting of DC motor, transmission and incremental encoder with resolution of 1 degree, which can be read out by “brick” through the output port which is connected to LEGO DC motor. DC motor characteristics are given in [9].

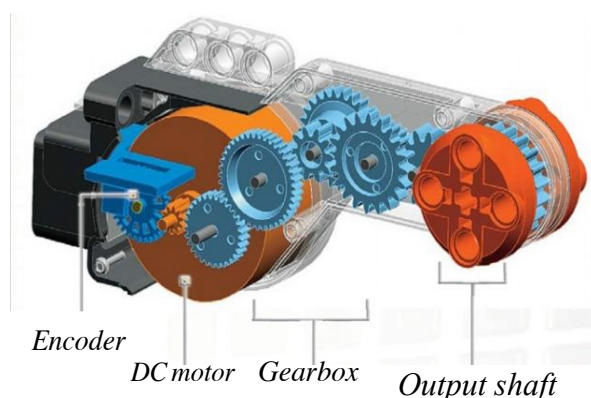


Fig. 2. LEGO DC motor with integrated encoder and gearbox

Nominal characteristics of LEGO DC motor are: 9V power supply, max rotational speed of 177 min^{-1} , torque of 16.7 Ncm, power of 2.03W, efficiency coefficient of 41%. Motors are driven by PWM signal, with desired direction and rotation speed in the range of -100 to 100, where sign defines direction and the absolute value is percent of the max motor power. Particular value of the motor power is kept in 8 bit register. Most significant bit stores sign information, while rest of the bits store information of the current motor power absolute value. If the motor power is greater than 127, the most significant bit becomes the logic 1, control unit interprets it as negative power and motor starts spinning in the opposite direction (robot starts moving backwards instead of forward). That should be taken care of in the program, when programming the robot movement by defining the numeric

value.

Described NXT set, provides for the robotic graphical programmable environment intended for users older than 10 years of age. For educational purposes on the higher levels, it is advisable to use some other means of programming available from Matlab, LabVIEW, NXC and similar. According to methodological approach applied in [10], we selected Python, as the open source high-level object oriented language. The two approaches are possible with Python. Program code in Python can be translated to NXC code which is compiled and stored in robots memory using the Bluetooth or the USB cable. In that way, the robot is autonomous after storing the program in memory. The other approach is to have the Python program code executed on PC, and executional commands sent to robot using USB cable or the Bluetooth. In that way, the more powerful data processing resources of the PC and advantages of the true object oriented programming are available, with constraint of robot motion being limited either by the USB cable length or within the Bluetooth range. In contrast with Matlab / Simulink support which besides basic usage licenses also requires even more expensive real time modules, the usage of Python is free and more useful from the educational view point. For tasks where the high level of programming skills are not of essential importance, the teaching staff should provide skeletal code as in [10]. Starting from the skeletal code, students are able to more efficiently devote to the task of primary educational focus.

III. EXPERIMENTAL SETUP AND OPTIONS FOR ON LINE EXPERIMENTING

The experimental platform consists of PC computer with Python programming environment, LEGO microcomputer (programmable unit, or „brick“ popularly) and two LEGO motors.

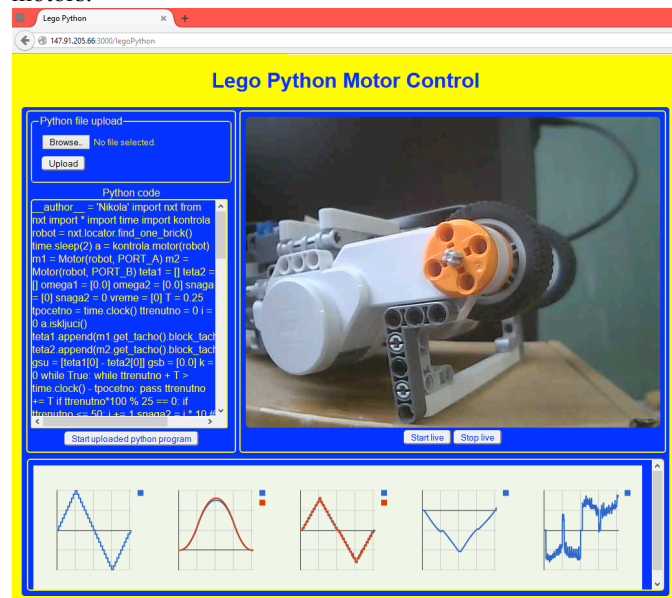


Fig. 3. Web page of the Web laboratory for remote control of Lego Mindstorms NXT DC motors (two motors are controllable)

Experimental task requires from student to drive both

LEGO DC motors with the same step function, record motor responses, perform identification of stationary states before and after input signal, perform parametric identification of dynamic model for transient process for both motors, and for adopted structure of the 1st order model (dynamic motor behavior)

$$G(s) = \frac{\omega(s)}{U(s)} = \frac{K}{T_m s + 1} \quad (1)$$

One example of experimental result is given in Fig 4 with normalized angle velocity obtained as real angular velocity, subtracted with its stationary value of angular velocity just before the step input signal. In the particular case of results from Fig 4, the difference in responses for the same input is obvious for two nominally same LEGO DC motors. Such a possibility requires special discussion with students.

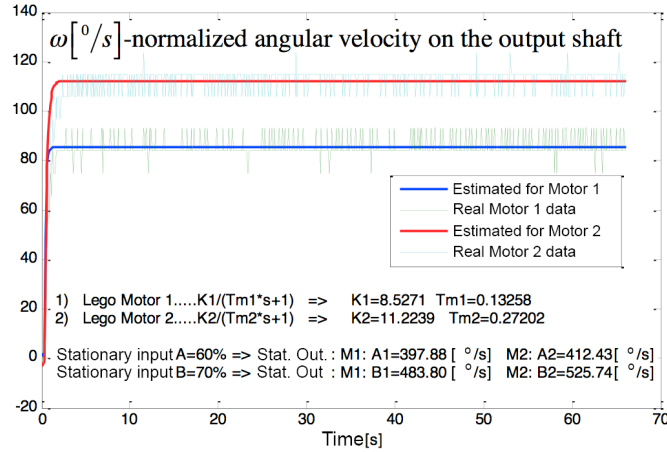


Fig. 4. Normalized dynamic responses of the LEGO DC motors of PWM step input $u(kT) = 60 + 10h(kT - 100T)$ [%].

The continuation of experimental task is similar to previous with the aim of determining and drawing of static characteristics and also determining of the nominal dynamic model of LEGO DC motor (one or more) which will be later used for design of servo controllers for position and velocity with LEGO DC motor acting as executive object. Responses for input upward and downward step function signals from the whole range of input signal values [-100%, 100%], and with step amplitude up to 5%, are recorded. In order to make the time duration of such an experiment shorter, it can be performed with more narrow input range

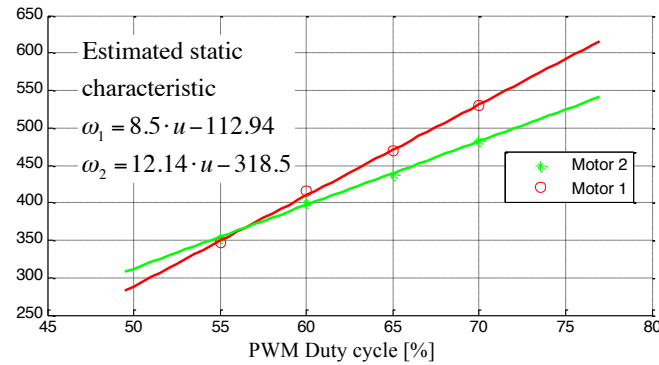


Fig. 5. Static characteristic of LEGO DC motor estimated from real data in range [55%, 70%] of the input signal

From series of experiments model parameters are calculated according to (1) for the each of LEGO motors. For case illustrated in Fig 5, 8 experimental trials were made with parameter identification using method of least squares that yielded model parameters (1) of motors 1 and 2 within ranges $K_1=[6.9, 9.9]$, $K_2=[10.5, 14.7]$, $T_{m1}=[0.023, 0.13]$, $T_{m2}=[0.07, 0.27]$, and average values $K_1=8.5$, $K_2=11.7$, $T_{m1}=0.07$, $T_{m2}=0.15$. From linearized static characteristics within the observed range, gains of $K_1=8.5$, $K_2=12.14$ are selected for nominal dynamic models of motors 1 and 2

$$G_1(s) = \frac{\omega_1(s)}{U(s)} = \frac{8.5}{0.11s + 1}, \quad G_2(s) = \frac{\omega_2(s)}{U(s)} = \frac{12.14}{0.11s + 1} \quad (2)$$

Time constants T_{m1} and T_{m2} are selected from identified ranges so that they are equal to sampling period in this particular case.

Experimental setup also enables the illustration of system frequency characteristics. For the identified dynamic model, the characteristic pass frequency is calculated and simulations as in Fig 6 are performed. Experimental verification of obtained simulation results is performed after that. The amplitude of the input signal should be adjusted to static characteristic (not to be out of linear region around work point selected for experiment). Input and output signals, phase frequency characteristics should be presented. Relationships between time and frequency domain together with frequency bandwidth, selectivity and experimental verification should be commented and explained.

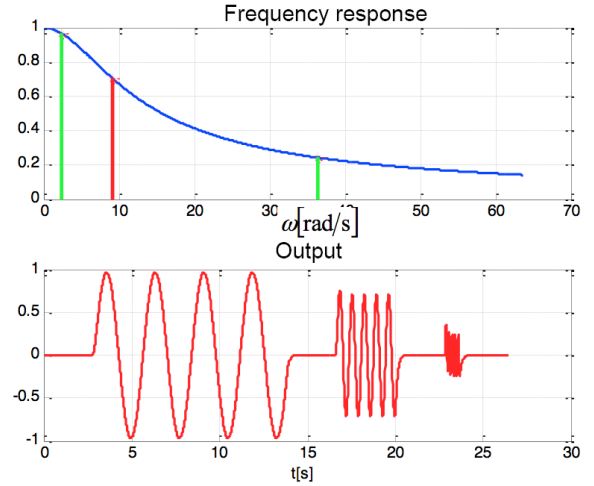


Fig. 6. Bandwidth of the LEGO DC motor. Below: time domain response to $1 \cdot \sin(\omega t)$, where $\omega = \omega_0/4$, $\omega_0 = 4 \cdot \omega_0$. Above: Frequency domain response to the same input with $\omega = \omega_0/4$, $\omega_0 = 4 \cdot \omega_0$ (ω_0 -bandwidth)

The next experimental task is synthesis of PI controller for velocity servomechanism with identified LEGO DC motors, so that the system transfer function in closed loop is: $G_{cl}(s) = 1/(2 \cdot \max(T_{m1}, T_{m2}) \cdot s + 1)$. As the dynamical models (2) differ within the area around selected work point in our case, the corresponding PI controllers will differ as well

$$G_{c1}(s) = T_{m1}/(2 \cdot K_1 \max(T_{m1}, T_{m2})) + 1/(2 \cdot K_1 \max(T_{m1}, T_{m2}) \cdot s), \quad (3)$$

$$G_{c2}(s) = T_{m2}/(2 \cdot K_2 \max(T_{m1}, T_{m2})) + 1/(2 \cdot K_2 \max(T_{m1}, T_{m2}) \cdot s).$$

Response of both (servo) motors should be recorded for the same input step function of reference value in the vicinity of

work point. In our case such an input for the system with closed loop can be:

$$r(kT) = 400 + 50h(kT - 100).$$

Motors angle velocities and their difference that is a measure of synchronization error should be recorded. Repeat the experiment for the case when: 1) disturbance is introduced on one of the motors (manually applying force to slow or stop the motor shaft), 2) Controller is transformed to P controller $G_{cl}(s) = K_{p1}$. Comment the error of stationary state, the quality of transient process and the error of synchronization for previously considered cases.

The last experimental task is continuation of a previous in the sense of application of the structure with cross coupling of two motors as shown in Fig 7 in order to minimize the synchronization errors of motors for velocity (e_{s_v}) and position (e_{s_p}). As in the previous case, both motors are driven by $r(kT) = 400 + 50h(kT - 100)$. Constant K_s in Fig 7 has value of 0.5 (to be changed from 0 do 1). Angular velocities ω_1 and ω_2 and reference value r are to be recorded on the same diagram. Synchronization errors are also recorded for: 1) nominal case, 2) on transformation of PI controller of the first servo motor to P controller, 3) in the case of disturbance (stopping one motor shaft by hand and observing if the other motor will also stop).

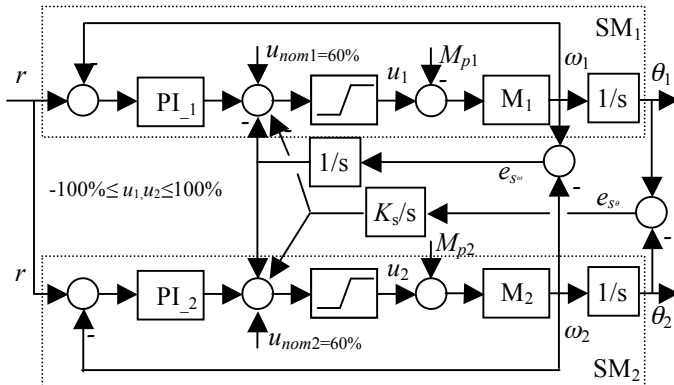


Fig. 7. Structure of system with cross-coupled servo motors SM₁ and SM₂ at set point ($u_{nom}=60\%$). M_{p1} and M_{p2} are external disturbance moments on process variable.

In this way, the robot being driven by two LEGO DC motors, which parameters can differ and change over time, and be subjected to various disturbances, can despite of all mentioned move in straight line at a given velocity (proportional to r).

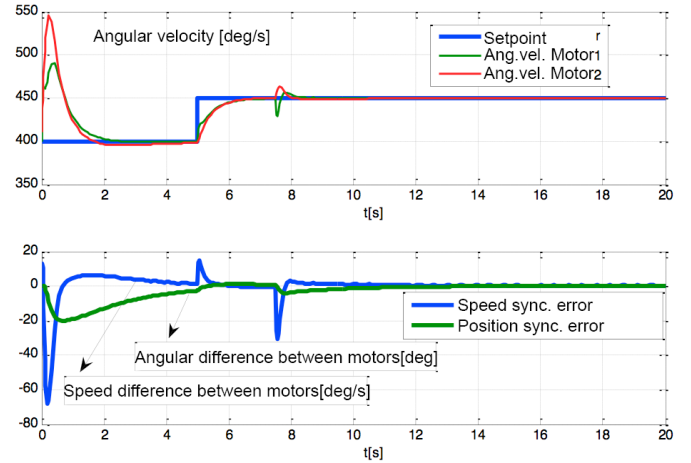


Fig. 8. Simulation of the system shown in Fig.7. Initial condition disturbance and disturbance $M_{p1}=M_{p2} = 5 \cdot h(t-7.5)$ are also shown. Parameters of M₁ and M₂ are different: $K_1=8.5$, $K_2=12.4$, $T_{m1}=0.0292$, $T_{m2}=0.2752$. PI gains (3) are same, except second PI regulator (PI₂) is reduced to P regulator (Integral term is turned off), for the robustness demonstration.

Various other experimental tasks are possible, like synthesis of velocity or positional servomechanism.

According to [10], the Python was chosen as free software that allows complete implementation of all previously mentioned experimental tasks. Besides being a free solution, it is also useful for engineering education. For quick and efficient introduction of students to Python programming language, a software tutor system was designed [13] with methodology and similar examples as in [10]. One of very important reasons for the choice of Python is a great number of existing modules with libraries of classes and functions for various applications that are free for use.

Python can be used in various ways through the web, from the client browser. In Fig 9 the basic concept of the LEGO web laboratory used for robot remote access and control, is shown.



Fig. 9. Basic concept of the LEGO Web Lab on University in Kragujevac

Remote user, a student, at internet-enabled location, using any up to date web browser can access LEGO web lab server. User login at laboratory server is mandatory. Registered users may be allowed to use LEGO web lab system in accordance with laboratory rules. Python programs that control robots are executed on PC in the laboratory, which is connected to LEGO robots with USB cable or Bluetooth radio. Logged in

student can upload a file with Python program that controls LEGO robot and test his program in that way. Remote user session uses AJAX for sending data to uploaded Python program. The effect of uploaded Python program execution can be observed through data that Python program generates and which are returned by web server. Web or IP cam enables additional visual monitoring of remotely controlled robot in the lab. Besides sending numerical data from web server as program results or robot visual monitoring, using only standard web client software technologies like HTML5, CSS and JavaScript it is possible with adequate animations, to represent effects of remote robot control, without requirements for browser plugins. Basic concept in Fig 9 is independent of computer operating system and hardware platform. Any web server can be used (Microsoft IIS, Apache, NodeJS Express...) on a classical PC, but it can be also quite modest configuration such as Raspberry Pi and LightTPD web server.

IV. NEW VERSION OF NXT PYTHON AND SKELETAL CODES

For LEGO Mindstorms NXT, NXT-Python package [7] was developed, which is a driver and program interface for robot programing. Also, some additional modules are required for robot control over USB port and Bluetooth radio from Python program. The Python itself can be used on various hardware–software platforms, which significantly expand potential for implementation of educational projects based on LEGO robots and Python. NXT-Python was successfully tested with various versions of Windows operating system – XP, 7, 8, 8.1, as well as Linux. But, using of available NXT Python libraries [7], NXT motor control results in responses as in Fig 10, which show the presence of “dead zone”, that would not be present if the same experiment is performed using Matlab/Simulink or the NXC.

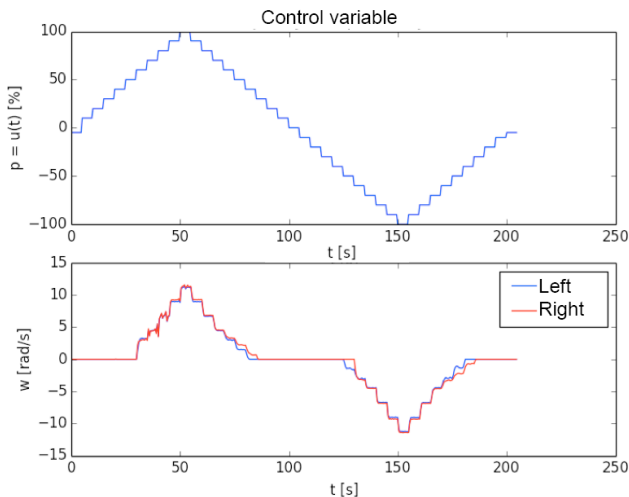


Fig. 10. Response of the NXT motors to multiple step input using available NXT Python library

Existing version of the NXT Python only approximately generates the PWM signal for driving the motor, as shown in Fig 11. There is no required amplitude of the PWM signal,

nor its expected shape, and the right part of the PWM signal does not exist.

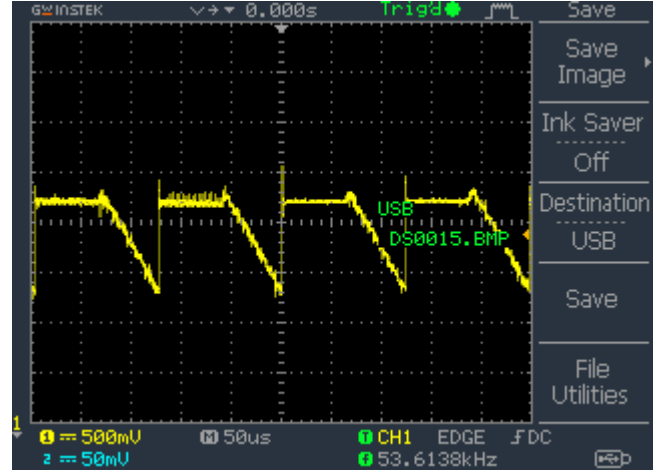


Fig. 11. PWM signal with 50% duty cycle obtained from existing NXT Python library

The problem was solved by integration of already existing program MotorControl [14, 2] with NXT Python [7]. MotorControl [14] is program written in NXC language (Not Exactly C). Its main purpose is to be the interface between LEGO robot and program package Matlab/Simulink. In the current version of NXT Python 2.2 [7] there is an interface between Python and program MotorControl, but it is poorly documented and complicated for use. Therefore, the wrapper for connection of MotorControl program and NXT Python library was designed and implemented. Description and installation details can be found at web page of University in Kragujevac [16]. Program MotorControl must be written to inner LEGO robot memory. Web page [16] contains program MotorControl, installation of the new NXT Python version, and skeleton files for typical laboratory exercises on experimental setup in Fig 3.

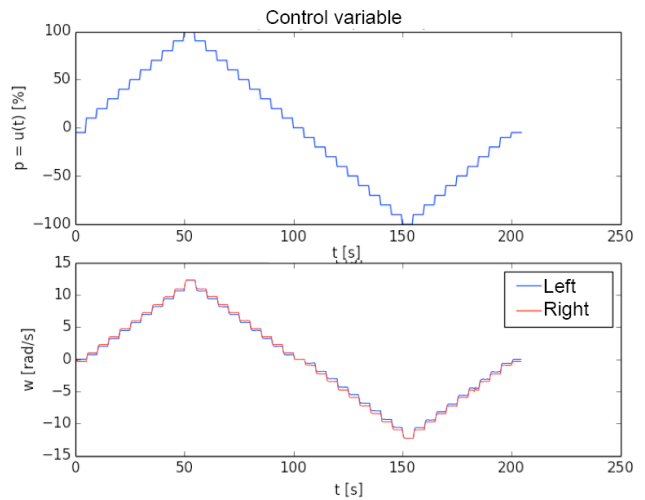


Fig. 12. Response of the NXT motors to multiple step input using new version of the NXT Python library

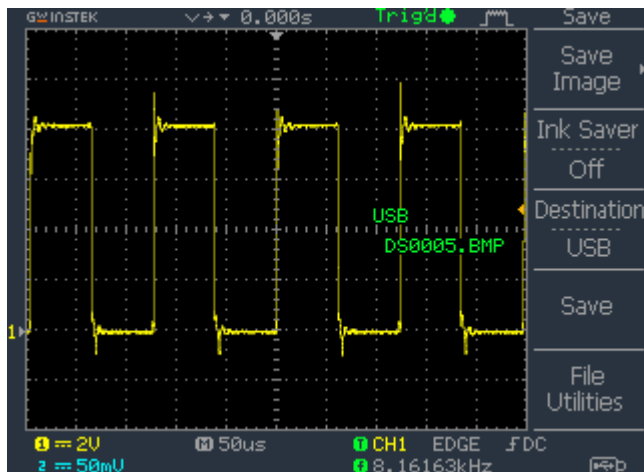


Fig. 13. PWM signal with 50% duty cycle obtained from new version of NXT Python library

For better availability and more efficient use of LEGO NXT resources, web page [16] was extended with functionalities of NXT Python WEB Laboratory, as shown in Fig 3. For a given configuration experimental setup, students can upload their Python code and start experiment execution. Students can develop Python code from scratch or can use and adapt existing skeleton model code.

V. CONCLUSION

This paper describes one aspect of application of LEGO Mindstorms NXT at University of Kragujevac. The new solution of NXT Python library was presented. It is freely available for all interested in parties together with developed software base of typical Python programs that help students with quick start of experimenting with LEGO platform without requirement of commercial software tools. WEB development environment was created for: 1) availability of information for software installation and performing of hands on experiments, 2) remote experimenting through the web which allows Python program file upload, execution in experimental environment and recording of execution results on real experimental setup.

This paper also contributes to mapping of educational methodology of world famous engineering schools to our local institutions.

ACKNOWLEDGEMENTS

The work on the paper was financially supported by the projects TR33022 and TR33047 financed by the Ministry for education, science and technological development of the Republic of Serbia.

REFERENCES

- [1] M.Matijević, Vladimir Cvjetković, Vojislav Filipović, Nikola Jović, "Basic Concepts of Automation and Mechatronics with LEGO MINDSTORMS NXT", *Tehnika*, Vol. 6, No 4, pp. 651-663, 2014.
- [2] Rising, James. *SP.293 Lego Robotics*, Spring 2007. (MIT OpenCourseWare: Massachusetts Institute of Technology), <http://ocw.mit.edu/courses/special-programs/sp-293-lego-robotics-spring-2007>
- [3] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Ballé, T. Herold, A. Telle, T.G. Noll, K. Hameyer and T. Aach, "MATLAB Meets LEGO Mindstorms - A Freshman Introduction Course Into Practical Engineering" *IEEE Transactions on Education*, vol. 53, no. 2, pp. 306—317 2010. <http://mindstorms.lfb.rwth-aachen.de/>
- [4] Massachusetts Institute of Technology, MIT 6.270 Course: Autonomous Robotics Competition, <http://scripts.mit.edu/~6.270/>, Lecture Notes 2010-2014: <http://scripts.mit.edu/~6.270/contestants/>, 2014.
- [5] B. H. Ferri, S. Ahmed, J. E. Michaels, E. Dean, C. Garyet, S. Shearman, "Signal processing experiments with the LEGO MINDSTORMS NXT kit for use in signals and systems courses", in *Proc. American Control Conference*, Hyatt Regency Riverfront, St. Louis, MO, USA, pp. 3787-3792, June 10-12, 2009.
- [6] Y.Kim, "Control systems Lab using a LEGO Mindstorms NXT Motor System," *IEEE Trans. on Education*, vol. 54, No 3, pp. 452-461, August 2011.
- [7] Marcus Wanner, *NXT Python*, <https://code.google.com/p/nxt-python/>, 2013.
- [8] R. Watanable, *NXT Motor Internals. NXT Motor Characteristics*: <http://www.philohome.com/nxtmotor/nxtmotor.htm>, 2007.
- [9] G.Sziebig, B.Takarics, P.Korondi, "Control of embedded systems via Internet," *IEEE Trans. on Education*, vol. 57, No 10, pp. 3324-3333, 2010.
- [10] K. Leslie, J. White, H. Abelson, D. Freeman, T. Lozano-Pérez, and I. Chuang. *6.01SC Introduction to Electrical Engineering and Computer Science I*, Spring 2011. (MIT OpenCourseWare: Massachusetts Institute of Technology), <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011>
- [11] M.L. Gasperi, P. Hurbain, *Extreme NXT: Extending the LEGO Mindstorms NXT to the Next Level*, second edition, Springer Verlag, New York, 2009.
- [12] Bonnie Heck, *LEGO NXT Experiments for Signals and Systems*: http://users.ece.gatech.edu/~bonnie/book3/LEGO_Experiments.html, Georgia Institute of Technology, Atlanta, 2014.
- [13] M.Matijević, V.Joković, *Uvod u objektno orijentisano programiranje – Elektronska zbirka zadataka sa softverskim rešenjem interaktivnog sistema testiranja podrške učenju*, CD izdanje projekta 144856-TEMPUS-2008-RS-JPGR, FIN Univerzitet u Kragujevcu, 2013: <http://cpa.fin.kg.ac.rs:30/Home/EPublishing?Length=4>
- [14] RWTH – Mindstorms NXT Toolbox for Matlab/Simulink, NXc program: *MotorControl*, <http://www.mindstorms.rwth-aachen.de/trac/wiki/MotorControl>, RWTH Aachen University, 2015.
- [15] M.Matijević, G.Jakupović, J.Car, *Računarski podržano merenje i upravljanje*, Univerzitet u Kragujevcu, 2008.
- [16] NXT web laboratory, <http://cpa.fin.kg.ac.rs/nxtlab.html>, 2015

Remote control of LEGO Mindstorms NXT Motors programmed in Python

Nikola Jović , Vladimir Cvjetković, Milan Matijević