Proceedings of the 2nd InternationalConferenceofControl,DynamicSystems,andRobotics
Ottawa, Ontario, Canada, May 7-8 2015
Paper No. 186

Multivariable Controller Design of a Lego Mindstorm NXT Robotic Arm

Victoria M. Serrano R., Michael Thompson, Konstantinos Tsakalis
Arizona State University
1151 S. Forest Ave, Tempe, Arizona
vmserran@asu.edu; mjthomp3@asu.edu; tsakalis@asu.edu

Abstract- In this paper we present a 3R Lego Mindstorm robotic arm for a multivariable controller design through the use of Matlab and Simulink. System identification was performed to estimate the parameters of the system for a pick and place task. A Proportional-Integral (PI) controller was designed for a decoupled plant. Tustin and Forward Euler discretization techniques were compared to the continous time controller to select the best method to be implemented for the Lego robot. These steps form the basic sequence of controller design and implementation. Their application to a widely accessible benchmark is a valuable example of control of a multivariable process, yet simple enough for an undergraduate audience. As an instructional paradigm, it provides a framework for a hands-on approach to control systems, blending some of the more advanced aspects of control theory with implementation details.

Keywords: Lego robotic arm, system identification, multivariable control design, Simulink.

1 Introduction

Multivariable control design has been a topic of great interest for a long period of time. Its application to industry has been widely extended due to the demand of more efficient and robust controllers; such devices are capable of handling systems with multiple inputs and multiple outputs (MIMO). Many of these controllers are tuned using extended versions of single-input single-output techniques. Among these, Proportional-integral-derivative (PID) controllers are still the most common type of controllers used in industry. With an extensive literature available on tuning and properties of PID controllers, they offer integral action to eliminate set-point errors and disturbance offsets, phase lead to adjust crossover properties like phase-margin –and, hence, closed-loop damping. At the same time, their simplicity allows for relatively straightforward implementation including discretization and ad-hoc, but very important, modifications for anti-windup, parameter scheduling. Moreover, their extensions to multivariable systems make them suitable in a plethora of applications.

Although MIMO plants are very common in industrial applications, their control is often performed in a decoupled fashion both for simplicity and integrity in case of failures. The interaction between loops can also be reduced through static or dynamic decoupling techniques. This procedure aims to make the system diagonal dominant and, therefore, easier to control. Katebi, 2012 studied some of these multivariable control methods and made a comparison in terms their stability and robustness performance. While this study contributed to a better understanding of multivariable controller design, the transfer of the technology to industrial or academic users of lower level of expertise is still an open issue.

The lack of hands-on experiences in some engineering courses has always been a challenge to overcome at universities. One of the main problems lies on the fact that equipment to develop experiments in courses such as control systems are usually expensive. This obstacle becomes even more crucial with the expansion of on-line programs, requiring either the use of remote-login technology for the use of laboratory facilities, or their undesirable conversion to simulation experiments. In this vein, we propose to use Lego Mindstorms kits to develop experiments that can be used to apply to control theory classes such as: system identification, controller design, discretization methods and multivariable control. Although some work has been done with Lego Mindstorms kits to teach engineering classes, some of the basic topics for a multivariable controller design has not been fully addressed.

Cruz-Martin et al., 2012 reported the use of Lego Mindstorms to teach data acquisition, control theory and real-time applications to undergraduate students. Similarly, Kim, 2011 used a Lego Mindstorms motor to teach control theory classes. Valera et al., 2011 also presented some work with Lego NXT based robots to design and implement Kalman filters. Behrens et al., 2010 have used these kits in addition to Matlab as an introductory course for an engineering class. Tse, 2009 presented the use of Labview to control Lego NXT-based robots. Kim et al., 2007 used visual programming to perform experiments with Lego Mindstorms NXT while Be et al., 2011 presented a wireless control Lego NXT robot using voice commands. However, all of them lack in applications on the control of multivariable systems.

This paper describes a method to design a multivariable controller for a 3R Lego Mindstorms robotic arm. A system identification process was used to obtain the plant model. The results of this exercise were compared to the first principle model of the system. A PI controller was designed and implemented using phase and gain margin specifications. Such controller was then discretized and implemented on the Lego robot. The compensated plant showed a good performance when the frequency response was analyzed, and the output followed closely the reference command, according to the design specifications.

2 Experimental Setup

A Lego Mindstorms NXT Base Set 9797 was utilized for the experiment. It contains hundreds of Lego pieces for construction, a Lego brick, four sensors, three motors and cables for connection. The Lego brick consists of a 32-bit ARM7 microprocessor with a 256 Kbytes of FLASH and 64 KBytes of RAM memory. Additionally, it allows USB and Bluetooth communication. It has four input ports to connect different sensors such as ultrasonic, light, accelerometer, gyroscope, GPS, touch and sound. It also allocates three output ports to connect 9 volts servo-motors equipped with optical built-in encoders. The brick powers with a battery pack which is equivalent to six AA batteries.

Furthermore, Matlab and Simulink were used for data collection. A version of Matlab 2014b was required to run the Lego Mindstorms NXT toolbox. A computer with incorporated Bluetooth was necessary to collect the data coming from the Lego brick. The Simulink model was downloaded to the Lego brick using a USB cable while the data was retrieved to the computer using Bluetooth communication.

3 Methods

Robotic arms are used not only in industry, but in many everyday tasks. Understanding how to model this physical system is an important first step to be able to design a controller and define meaningful specifications for its tuning. First principles can be used to provide an idea of how a system behaves under specific conditions. However, the details of modeling to accurately match the observed response may become too tedious. One approach to alleviate this problem is to employ a system identification procedure to estimate the effective parameters of a robotic arm. The Lego Mindstorms NXT kit includes servo motors with built-in encoders that can be used to estimate such parameters. By using Matlab and Simulink and introducing a specific input to each motor, their corresponding angular rotation can be retrieved at the workspace. The input was downloaded to the brick using a USB cable while the output was collected through Bluetooth communication. Both inputs and outputs were used to identify the system. Then the identified system was compared to the model specified by first principles.

3.1 Plant Modeling Using First Principles

The plant modeling stage involves a series of steps such as first-principles analysis, system excitation, parameter estimation and uncertainty estimation. Using first-principles, one can infere the mathematical description of the system. This is crucial for the system identification process, since it provides at least a rough idea of the underlying model structure. By having a prior description of the system, we can design the excitation to provide sufficient information in the frequencies of interest, e.g., a decade around the intended gain crossover frequency. Additionally, the uncertainty estimation enables us to determine suitable performance specifications and the reliability of the identified model for controller design.

Once the first-principles dynamics have been determined, we can define what input signal should be generated in order to identify the system properly. According to Kim, 2011, a DC Lego motor can be modeled as (1). Additionally, a gear train system can be described as (2) assuming that there is no inertia, no losses and the system is perfectly rigid. This configuration is consistent with the robotic arm used for the experiment. Furthermore, the motor that controls the up and down movement of the robotic arm can be modeled as an inverted pendulum.

Therefore, by applying first principles and assuming that there is no coupling between different inputs and outputs, the robotic arm can be approximately described by Equations (3) and (4).

$$\frac{\Theta(s)}{V(s)} = \frac{k}{s} \cdot \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

$$\frac{\Theta_C(s)}{\Theta(s)} = \frac{r_1}{r_2}; \frac{\Theta_B(s)}{\Theta(s)} = \frac{r_1}{r_2} \tag{2}$$

$$\frac{\Theta_C(s)}{V(s)} = \frac{k}{s} \cdot \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{r_1}{r_2} \tag{3}$$

$$\frac{\Theta_B(s)}{\tau(s)} = \frac{\frac{1}{mL^2}}{s^2 + g/L} \tag{4}$$

Here we assumed that the inverted pendulum can be modeled as a massless rod of length L with a point mass m attached at the end. The gravitational acceleration is denoted by g. Θ represents the angular position of the motor, V is the voltage applied to the motor, k is the system gain, ζ the damping ratio and ω_n represents the natural frequency. According to [**?**], k is 8.34, ζ may be assumed to be equal to 1 and ω_n equals 40. Θ_B and Θ_C represent the angular position of motor B and C, respectively. Additionally, the torque and the power level for an NXT Lego motor can be obtained from [**?**].

The gear system for each motor is given in Table 1.

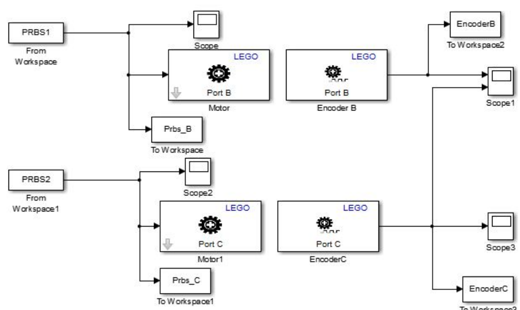
Table 1: Gear system for the DC motors

Radius	Motor B (cm)	Motor C (cm)
r_1	0.48	0.48
r_2	2.08	2.90

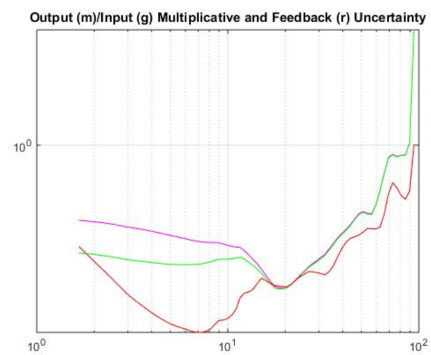
3.2 System Excitation and Parameter Estimation

A pseudo random binary sequence (PRBS) was used to excite the motors of the robotic arm. The excitation should have sufficient energy around the desired closed-loop bandwidth. According to Tsakalis et al., 1997, the signal should be large enough, so that the signal to noise ratio is good, but small enough for the system to be approximately linear around the operating point. The Fast Fourier Transform (FFT) technique was used to assess the level of excitation within the frequencies of interest.

A parametric system identification process Ljung, 1999 was performed to estimate the parameters of the robotic arm at the upright position. Since this is a multivariable system, the identification will also reflect the coupling between different inputs and outputs. Two independent inputs were generated and applied to each motor of the arm as shown in Fig. 1a. The inputs correspond to the power applied to each motor that can vary in a range from -100 to 100. The outputs provide information of the angular position in degrees of each motor axle. Motor C will be controlling the rotation of the link while motor B controls its vertical position. Due to physical limitations motor B could only rotate in a range between -430 to 315 degrees from the link's upright position while motor C can rotate from -1000 to 850 degrees from a position that is perpendicular to the Lego brick.



(a) Simulink model for a Multivariable System

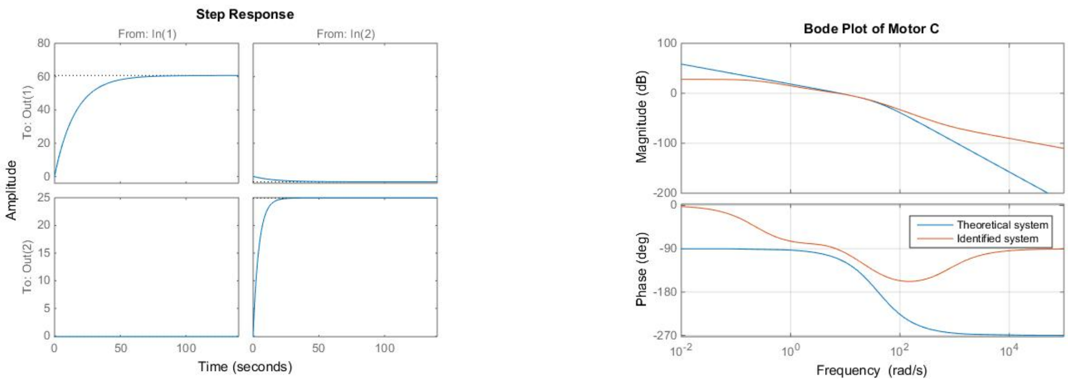


(b) Uncertainty bounds

Fig. 1: System Identification of a Lego Robotic Arm

A state-space representation was obtained for the multivariable system. Fig. 1b shows the uncertainty bound for the multivariable system. This was determined by computing the frequency content of the residual errors and the frequency content of the plant input and output. Its usefulness is that it allows us to determine the confidence in the parameter estimation in terms of a control objective, instead of the much more vague "energy of fitting error" or other statistical fitting metrics. In a loose description, the uncertainty estimation amounts to the normalization of the energy of the residuals with the energy of the excitation and their translation as an effective "multiplicative uncertainty". For such forms of dynamic uncertainty, the "Small Gain Theorem" establishes bounds for the closed-loop sensitivities in order to guarantee stability of the perturbed closed-loop system. Its implication is that the data fitting residuals can now be translated into a dimensionless "robust stability condition" that quantifies the confidence in the identified model for the design of a controller with a given set of specifications (typically, the loop crossover frequency or closed-loop bandwidth). Alternatively, the same data can also be used to define control specifications for which the model can lead to a reliably predictable closed-loop system. (For more details, the reader is referred to Tsakalis et al., 1997, Tsakalis et al., 2002.)

Additionally, a step response was plotted to observe the coupling or decoupling between different inputs and outputs. After observing Fig. 2a, we may highlight the following results:



(a) Step Response of the Multivariable System.

(b) Theoretical vs. identified system for the Lego motor C

Fig. 2: Identification of the System

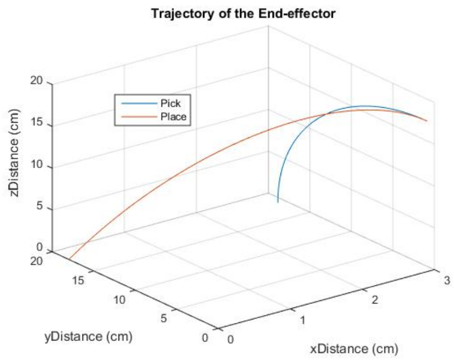
- The step response from PWM (In1) of motor B depicts a significant value for the angular position (Out1) of motor B.
- The step response from PWM (In2) of motor C also depicts a significant value for the angular position (Out2) of motor C.
- PWM of motor C does not show a significant interaction on the angular position of motor B.
- PWM of motor B neither affects the performance of the angular position on motor C.
- The step response of this representation shows that the system is diagonally dominant. Thus, the interaction between applied power and angular position for each of the motors is approximately independent. Therefore, this multivariable system can be treated as two single-input single-output (SISO) plants for controller design purposes.

4 Trajectory Planning

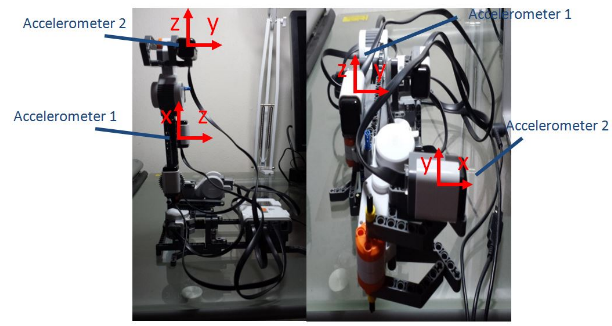
Two states were considered for trajectory planning purposes: angular position and angular velocity. According to Thompson et al., 2014, the workspace and the velocity manipulability that the robotic arm can reach may be described by ellipsoids. Therefore, having information of angular position and angular velocity allowed us to define the initial and final states of the system to determine the optimal path Naidu, 2003 as shown in Fig. 3a. With that purpose, a set of two accelerometers were placed at the claw and on the link to compare the angular velocity reading to the gyroscope values. The gyroscope is capable of measuring the angular velocity about one axis at a time in a range that goes from -360 to 360 degrees/sec. By implementing two accelerometers separated by a distance "D", one can infer the value of the angular velocity. This is obtained through manipulation of the acceleration data on the x, y and z axes of each accelerometer. By default, the configuration of the accelerometer for the gravity vector has the z-axis pointing upward from the ground. This corresponds to a value of approximately 200 counts on the Lego sensor. The position and direction of each sensor is shown in Fig. 3b.

5 Controller Design

We used Matlab and Simulink to design and implement the controllers for the decoupled plant. A proportional-integral (PI) controller was designed using frequency response specifications. The phase margin was choosen to be approximately 50 degrees and the crossover frequency equal to 15 rad/sec. This value



(a) Planned Motion of the End-Effector



(b) Location and directionality of the two accelerometers

Fig. 3: Trajectory Planning

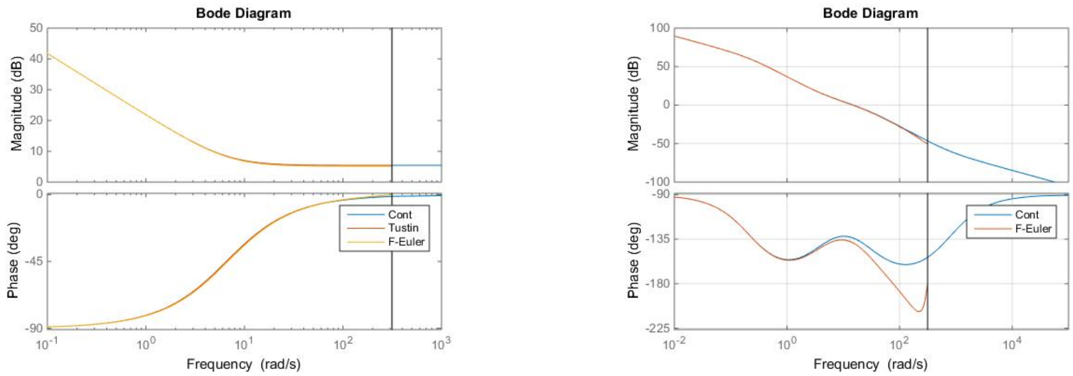
has ample margin from the upper limit suggested by frequency where the multiplicative uncertainty exceeds unity, which is 70 rad/sec. The robotic arm consisted of two motors that control the position of the end effector and another motor to grasp the object. However, for controller design purposes the effect of the latter is not considered. Additionally, since the system is decoupled, the rotation of every motor will not affect the performance of the other one. Therefore, each controller can be designed independently.

After the controller was designed for the continuous system, the system and the plant were discretized using a sampling time of 0.01 seconds. For this case, the discretization Zero Order Hold (ZOH) contributes only 4 deg. of phase lag and ignoring it does not cause severe deterioration of the loop properties. If this is not acceptable, one could augment the plant by a half-sample delay (approximating the ZOH) and design the controller to achieve the desired phase margin at the crossover frequency. Next, the discretization of the controller can be performed by different techniques, each having its own pros and cons. Fig. 4a shows a comparison of two different discretization methods Tustin and Forward Euler. Tustin provides a better approximation of the continuous-time controller phase, and is the predominantly preferred discretization method. On the other hand, Forward Euler is simpler and preserves the controller strict causality, but must be used with caution when the controller contains fast poles, as it may be the case with a PID controller (the sampling frequency must be faster than the pole frequency to preserve stability of the discretization). Here, the sampling rate is significantly faster than the crossover frequency and the both responses are very similar. As modern hardware allows for high sampling rates, the controller discretization is usually not presenting a problem. If this is not the case, e.g., due to sensor limitations, then additional phase lead must be introduced; in more extreme cases, the design may need to be performed entirely in discrete time.

Finally, it is always a good practice to use anti-windup modifications for the implementation of the controller (Astrom et al, 2005). Practical actuators always have saturation limits and these must be used in the controller algorithm to avoid integrator windup that could result to poor behavior and even instability. Effective anti-windup modifications for PID's can be as simple as using limited integrators in their implementation, a commonly available block in SIMULINK.

6 Results

After utilizing the PRBS signal for the system modeling, we designed and implemented the controller for the multivariable system. As shown in section 3.2, the system is diagonal dominant. Therefore, the plant may be treated as it would have two completely independent transfer functions for controller design purposes. Fig. 2b shows the frequency response of the analytical and the identified system for the Lego motor C.



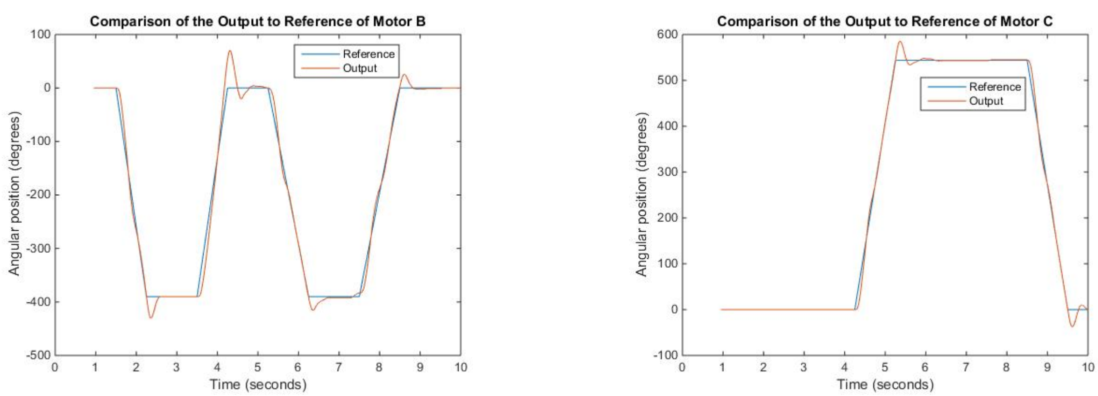
(a) Continuous vs. discretized controllers

(b) Compensated loop using a PI controller

Fig. 4: Discretization of the System

Once the controller was designed in continuous time and discretized using the Forward Euler method, the frequency response of the compensated plant was compared to determine if the controller met the specifications. Fig. 4b shows a comparison of the compensated loop of the continuous versus the discretized system. Although the magnitude and phase responses of the system start deviating at higher frequencies, the discrete response closely approximates the continuous-time system around 15 rad/sec.

Finally, the discretized controller was implemented in the Lego robot to follow a trajectory. Fig. 5a and Fig. 5b depicts the tracking reference for motor B and motor C, respectively. Although there is a small overshoot for each transition of the trajectory, the settling time occurs in less than 0.5 seconds, consistent with our closed-loop bandwidth specification.



(a) Tracking reference of motor B.

(b) Tracking reference of motor C.

Fig. 5: Tracking reference of different motors for the trajectory of the robotic arm.

7 Conclusion

In this paper we have discussed the modeling, design, and implementation of a multivariable controller for a Lego Mindstorms NXT robotic arm. A system identification process was performed to obtain the transfer function of the decoupled plant. By using affordable equipment, we could build and program a robotic arm which exhibits a similar configuration with modern practical systems. The approach is simple enough to be suitable for instruction at an undergraduate level and on-line programs. While the plant considered here does not show significant interaction between channels, and, consequently, the multivariable design was not very challenging, the approach is general and can be applied to more complicated cases. However, the study of more difficult multivariable systems in an educational environment through carefully selected experiments still needs to be addressed. Further topics of interest may include switching control or gain scheduling, motivated by the problem of the robotic arm grasping payloads with different weights.

References

R. Katebi, Robust Multivariable Tuning Methods, PID Control in the Third Millennium, Advances in Industrial Control, 2012, pp. 255-280.

A. Cruz-Martin et al., A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses, Computers and Education 59, 2012, pp. 974-988.

Y. Kim, Control Systems Lab Using a LEGO Mindstorms NXT Motor System, IEEE Trans. on Education, vol. 54, no. 3, Aug. 2011, pp. 452-461.

A. Valera et al., Design and Implementation of Kalman Filters applied to Lego NXT based Robots, Preprints of the 18th IFAC World Congress, Italy, Aug.-Sept 2011, pp. 9830-9835.

Behrens et al., MATLAB Meets LEGO Mindstorms-A Freshman Introduction Course Into Practical Engineering, IEEE Trans. on Education, vol. 53, no. 2, May 2010, pp. 306-317.

S. Tse, MINDSTORMS Controls Toolkit: Hands-On, Project-Based Learning of Controls, M.S. thesis, School of Engineering, Tufts Univ., Medford, Massachusetts, 2009.

S. Kim and J. Jeon, Programming LEGO Mindstorms NXT with visual programming, Int. Conference on Control, Automation and Systems 2007, Seoul, Korea, Oct. 2007, pp. 2468-2472.

D. Be et al., Wireless Control LEGO NXT robot using voice commands, Int. Journal on Computer Science and Engineering, vol. 3, no. 8, Aug. 2011, pp. 2926-2934.

K. Tsakalis and K. Stoddard, Integrated Identification and Control for Diffusion/CVD Furnaces, 6th IEEE Int. Conference on Emerging Technologies and Factory Automation, Los Angeles, CA, Sept. 1997, pp. 514 - 519.

L. Ljung, System Identification, Theory for the User. New Jersey: Prentice Hall, 1999.

K. Tsakalis, S. Dash, A. Green and W. MacArthur, Loop-Shaping Controller Design From InputOutput Data: Application to a Paper Machine Simulator, IEEE Trans. on Control Systems Technology, vol. 10, no. 1, Jan. 2002, pp. 127-136.

M. Thompson, V. Serrano, J. Willem and V. Martinez, Learning Robotic Concepts with a 3R Lego NXT Robotic Arm, Journal of Automation and Control Research, vol. 1, year 2014, pp. 38-44.

D. S. Naidu, Optimal Control Systems. Florida: CRC Press, 2003.

K. Astrom, T. Hagglund, Advanced PID Control. NC: The Instrumentation, Systems, and Automation Society; Research Triangle Park, 2005.

Web-1: http://www.philohome.com/nxtmotor/nxtmotor.htm, consulted 5 Jan. 2015.