122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society

Paper ID #11981

Using Simulink, Matlab, and LEGO Mindstorms to teach a Project-Based Control Systems Design Course

Dr. Estelle M Eke, California State University, Sacramento

Estelle Eke is a full professor of Mechanical Engineering at California State University, Sacramento. She received a B.S. degree in Aeronautical and Astronautical Engineering from Purdue University, a M.S. in Mechanical Engineering and Materials Science from Rice University, and a Ph.D. in Aeronautical and Astronautical Engineering from Rice University. She worked for two and half years in the Spacecraft Navigation Section at the Jet Propulsion Laboratory in Pasadena, and then taught for two and half years in the Department of Aerospace Science Engineering at Tuskegee University before joining California State University, Sacramento. While at Tuskegee University, she received the Teacher of the Year award in Aerospace Engineering for two consecutive years. At Sacramento State, she was named Outstanding Teacher in the College of Engineering and Computer Science in 2000. She teaches courses in the areas of Computer Applications in Engineering, Dynamics, and Controls. Her research interests are in optimization and robotics. She also serves as a design judge for FIRST Robotics competitions at the elementary and high school levels.

Using Simulink, Matlab, and LEGO Mindstorms to teach a Project-Based Control Systems Design Course

Abstract

Teaching control systems design using theoretical design examples outlined in most textbooks has been found to be quite challenging for many engineering students. One major observation is the tendency for students to resort to a trial and error approach in the design process without realization of the adverse effects that such an approach can have in the real world. Computer simulations of analytical solutions do provide some insight but here limitations arise in interpreting the results correctly or in the visualization of the real system's dynamics. This paper gives an overview of an approach that (i) allows the student to create and analyze the mathematical model of a real system, (ii) build a practical emulation of the real system using LEGO Mindstorms, (iii) observe the behavior of the system in real-time, and (iv) apply feedback principles using Matlab and Simulink to design the system to meet desired specifications. The first part of the course is an introduction to control theory using the classical approach. In the second part, students work in teams of three or four to design, build, and program a LEGO Mindstorms system of their choice that must perform stipulated tasks. The original system must not have any type of controller and must show that the desired requirements are not met. This provides an opportunity for the students to see how design constraints are established. Based on my observations so far, as well as on student feedback, students become very highly motivated by the design component of the course, and many end up with a thorough grasp of the fundamental principles of control systems design. The open-ended approach allows for creativity and flexibility in the design process, with both the instructor and students benefitting from an array of designs. A few examples of actual student projects will be presented.

Motivation

The incorporation of project-based learning (PBL) in the curriculum has been known to help students develop an intuitive understanding of the theory by providing real world applications that foster research and design. Bernard M. Gordon [1] presented a review of several institutions in the United Kingdom and Australia that adopted PBL in different types of engineering courses, and the positive impact the approach had on assessment. Students were encouraged to work in groups and document their progress throughout the process. Fernandez-Samaca et al[2] designed an undergraduate electrical engineering control system course using PBL. A series of courses that had both lecture and laboratory components were offered with the project being the central element in the approach. Enikov et al[3] developed the *Aeropendulum Project* which is a low-cost hands-on experiment suitable for a classical controls course. After using this project for a few years in the controls course at our institution, a missing component was the flexibility that would allow testing of other designs in addition to the pendulum. Golnaraghi and Kuo[4] developed *The Control Lab* which consists of virtual lab control experiments. Matlab and Simulink are used extensively in simulations and controller design projects. However, in order to run the simulations a special applet, the Automatic Control Systems (ACSYS), must be used.

Mathworks provides webinars that introduce control system design and analysis in Simulink. Examples using the *Matlab control systems toolbox* to design, tune, and implement controllers allow for quick and efficient development of real systems. Noting that the controls course at our institution does not have a laboratory component, and in order to use Matlab and Simulink as the design tool, a deliberate choice was made to combine the theory covered in lectures with a project-based learning approach. This bypasses the need to use specialized software such as ACSYS.

Seniors in the Department of Mechanical Engineering are required to take an automatic controls course. The prerequisite courses are Dynamics, Introduction to Technical Problem Solving, and Modeling and Simulation of Mechatronics and Control Systems. In the technical problem solving course students use Matlab and are introduced to programming a Parallax robot[5] using a lower-level PBasic language which provides an opportunity for hands-on applications with physical systems. This opportunity does not exist in the other two prerequisite courses. With the realization of the tremendous benefits that students gain using the project-based approach in the technical problem solving course, it was deemed appropriate to use robots in the controls course as a means for establishing a link between theory and application.

Teaching classical automatic controls over the years, it was noted that some students are intimidated by the theory. Mathematical concepts learned in the first two years in our engineering curriculum must not only be mastered but applied; equations are memorized but there is a disconnection when it comes to applying the equations in solving design problems. Most controls textbooks provide a set of design problems that are meant to help reinforce the principles but for the majority of students it still remains a challenge. The text used in this course is *Controls Systems Engineering* by Nise [6]. During the search for a robotics kit that can be used in demonstrating control concepts, emphasis was placed on (i) affordability: the robotics kit should be low-cost, and provided by the department (ii) software: the Student Edition of Matlab and Simulink which students are already familiar with in the programming course should be used, and (iii) flexibility: different designs should be implemented with ease allowing for a wide variety of open-ended projects. MathWorks Inc. developed Simulink Support Packages for hardware such as the Arduino and the LEGO Mindstorms NXT Hardware[7]. A deliberate choice was made to use LEGO Mindstorms NXT (NXT) so as to bypass circuitry design, and focus on programming, especially as a lab component is not included in the controls course. In the spring 2014 semester the project-based approach using the NXT was introduced to students.

Course Structure

The syllabus is structured so as to cover all items listed in the learning outcomes, which state that the students should be able to:

i. Use the terminology necessary to define a control system, distinguish between open-loop and closed-loop systems, and state the advantages of control systems.
ii. Find the transfer function of a system.
iii. Find a mathematical model, in state-space representation, for a linear time-invariant (LTI) system.
iv. Find the time response from the transfer function, and analyze the time response of first-, second-, and higher order systems.

v. Determine the stability of a system.
vi. Sketch a root locus and design a feedback control system to meet performance specifications via the root locus.
vii. Apply frequency response techniques in system design.
viii. Understand real devices that require modeling, simulation and control systems analysis and design.
ix. Use software to accomplish optimized designs.
x. Professionally document procedure and results of the final project in a technical report.

Resources on the internet that deal with the modeling of the NXT DC motor using Simulink[8,9], and Matlab tutorials on the root locus method[10] compliment the lectures. The Root locus technique is the primary design method used in projects to date. The primary mode of delivery consists of two 75-minute lectures per week. The projects are assigned around week 8 of the regular 16-week semester and students work in three or four-member groups. Class sizes range from twenty to forty students. The department supplies each group with an NXT kit which must be returned at the end of the semester.

Matlab and Simulink are used throughout the semester. After covering topics (i) through (v), root locus sketches are discussed followed by an introduction to proportional (P), proportional derivative (PD) and proportional integral derivative (PID) controllers. In the next phase where the design via root locus is implemented, Matlab's graphical control design tools, *rltool* and *sisotool*, are used. The ability to verify hand-computations using these tools help to reinforce the theory. Upon setting the design requirements settling time, percent overshoot, damping ratio or natural frequency on the root locus plots, an acceptable region for controller design is identified. Students must understand how to make a judicious choice of parameters to satisfy system constraints. In the practical environment when working with the NXT, the values of the gains of the different controllers generated with *rltool* or *sisotool* simulations may still need to be fine-tuned in order to get the ideal response in real time. In this situation the PID tuner in Simulink is invoked. The GUI provided by the PID tuner allows for convenient iterations by manually changing the gains, or the tuner can be made to automatically calculate the gains. Upon accepting the gains the NXT model is controlled in real-time to see how well the results meet expectations. Students are encouraged to design simple systems which do not require the use of system identification techniques as such a course is not offered in the undergraduate curriculum, and the necessary equipment is not available in our department. Nonlinearities such as saturation at high input voltages, dead zone at low input voltages, and in some instances backlash due to the gears becoming loose in a gear train are discussed.

Student Projects

Phase 1-The uncompensated System:

Students analyze the uncompensated system using the transfer function for the LEGO Mindstorms DC Motor [8] and shown in Figure 1.

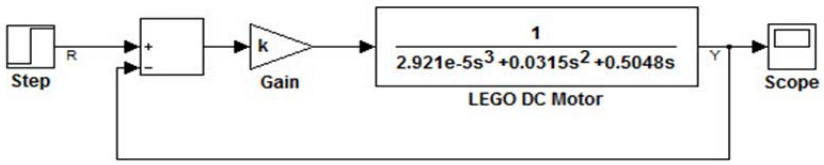


Figure 1. Uncompensated Block Diagram of LEGO DC Motor

The plant, $G_p(s) = \frac{1}{2.921(10^{-5})s^3 + .0315\, s^2 + .5048s}$ shows that the DC Motor is a Type 1 system.

The steady-state error is zero in response to a step input. The range of k for stability is obtained from the Routh criterion. The closed-loop transfer function is

$$T(s) = \frac{kG_p(s)}{1 + kG_p(s)} = \frac{k}{2.921(10^{-5})s^3 +\ .0315\, s^2 +\ .5048s + k}$$

Table 1. Routh Array

s^3	$2.921(10^{-5})$	0.5048
s^2	0.0315	k
s^1	$0.0159 - 2.921(10^{-5})k$	0
s^0	k	0

From Table 1, the range of k for stability is $0 < k < 544.3$.

Students sketch the root locus, and verify the sketch using Matlab's *rltool* (Figure 2), or *sisotool*. The step response (Figure 3) is also generated and the system parameters such as rise time and settling time are noted. Phase 2 addresses the design of the controller.

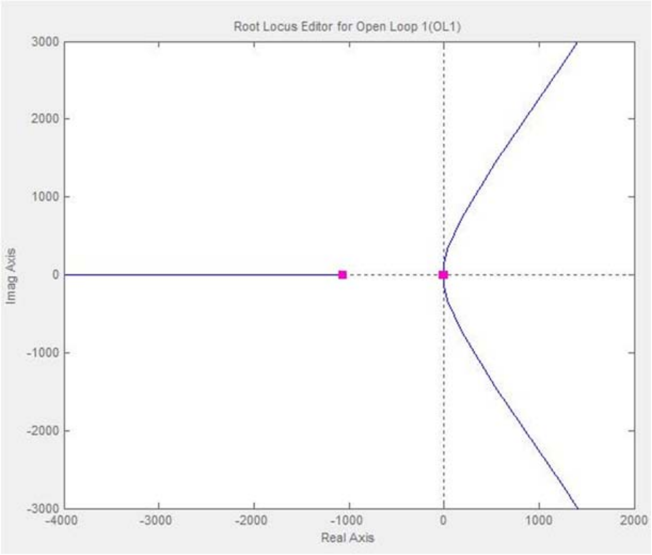


Figure 2. Root Locus of Uncompensated System via rltool

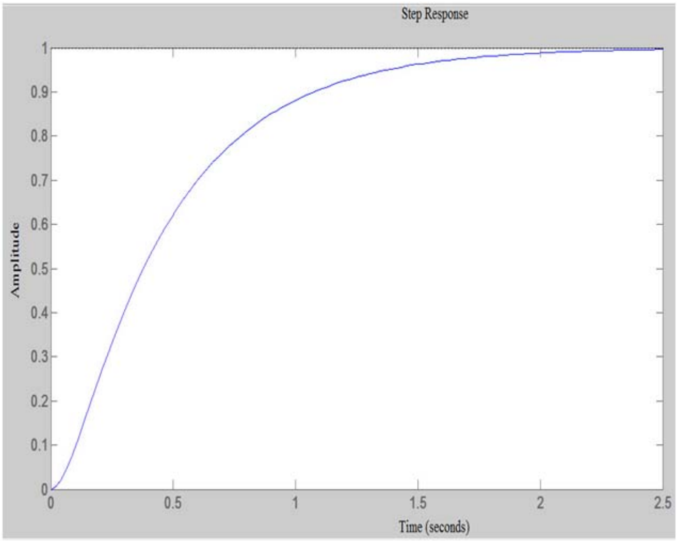


Figure 3. Step Response of the Uncompensated System

Phase 2-The Compensated System: Four projects are now presented and videos of the projects showing that the design specifications are met are available on youtube.

i. Project 1: Summer 2014[11]: Rope Climber and Ball Drop using Proportional Control.
ii. Project 2: Summer 2014[12]: Black Line Follower via Proportional Derivative Control.
iii. Project 3: Fall 2014[13]: Pick and Place a ball using Proportional Control.
iv. Project 4: Fall 2014[14]: Perform Color Detection, Grab and Drop a Red Ball via Proportional Derivative Control.

Project 1: Rope Climber and Ball Drop using Proportional Control

Objective**:** To program the NXT robot to climb up a diagonally mounted rope, stop fifteen centimeters from a backboard located at the end of the rope, and drop the ball into a miniature sized basketball hoop. All three motors provided in the NXT kit will be used in the design. The ultrasonic sensor will produce a signal to start rotating designated motors. Two of the motors will transport the robot along the string until it reached the specified distance from the backboard. At the end of the translational motion the third motor will be used to drop the ball into the hoop by performing a 360 degree rotation.

Constraints to be satisfied: The uncompensated system did not accomplish the objective because it had a variance in the stopping distance from the target. This caused an untimely initiation of the ball dropping sequence. To improve the accuracy and consistency of the stopping distance a compensator had to be added to the system. A proportional controller, k, was selected based on the “acceptable zone” on the root locus. The point at which a zeta-line that corresponds to 5% overshoot intersects the root locus was the design point selected; the value of the gain, $k = 4.24$.

Compensated System: After zooming in on the root locus a close-up view of the design point at 5% overshoot is shown (Figure 4.)

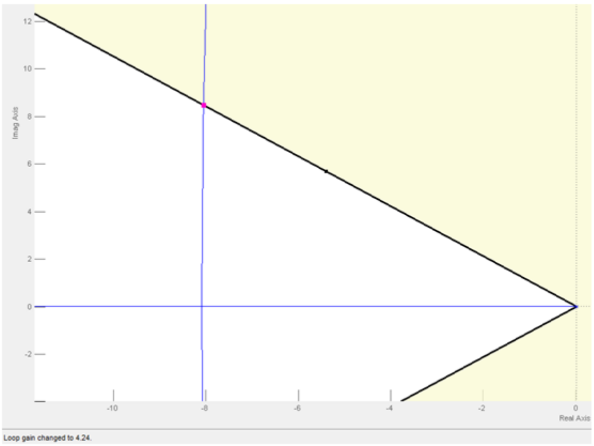


Figure 4. Root Locus showing design point at 5% Overshoot.

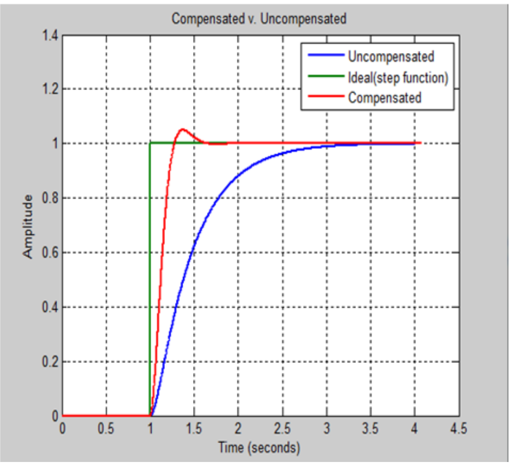


Figure 5. Time Response Plots

Time response plots for the uncompensated and compensated systems are shown in Figure 5. The NXT model of the compensated system is shown in Figure 6. In the uncompensated system the gains KA, KB, and KC were each set to unity.

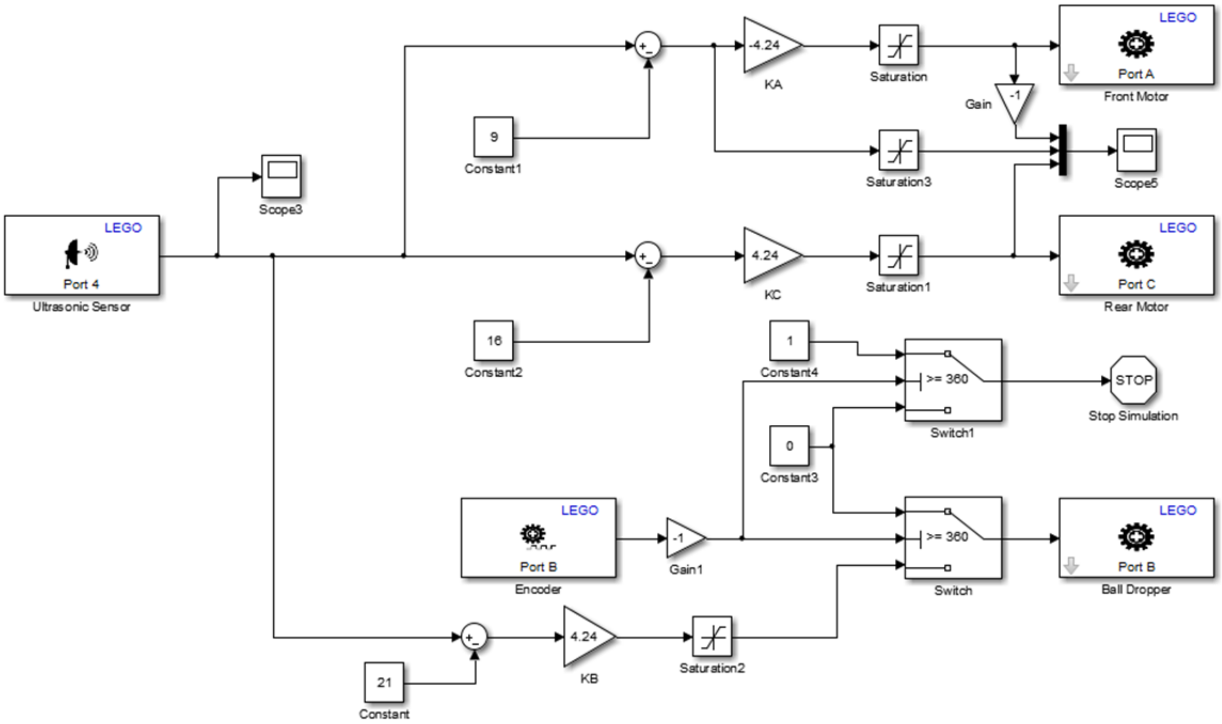


Figure 6. LEGO NXT Model for Compensated Climber

Table 2. Predicted Characteristics of Compensated Proportional Control System

	Compensated
Plant and Compensator	$\frac{k}{.00002921s^3 + .0315s^2 + .5048s}$
k	4.243
Dominant Poles	-8.07 ± j8.46
ζ	0.69
ω_n	11.7
%OS	5%
T_r (sec)	0.179
T_s (sec)	0.514
T_p (sec)	0.371

Figure 7. Uncompensated System shows translational motion incomplete.

Figure 8. Compensated System shows the ball being dropped into a hoop at end of the climb

Summary: The robot in the uncompensated system stopped before reaching the hoop and failed to deliver the ball (Figure 7.) In the compensated case, after completing the climb, an accurate delivery of the ball was accomplished (Figure 8.) Design specifications were satisfied.

Project 2: Black Line Follower via PD Control

Objective**:** To program the NXT robot to smoothly follow a black line. One light sensor will be used to detect the amount of light that indicates the robot's position with respect to the black line.

Constraints to be satisfied: In the uncompensated system the robot exhibited jerky movements and oscillating turns that forced it to veer off the black line. The focus will be on adjusting the transient characteristics of the system. The specifications to be satisfied are

%OS = 4.5% and $T_{s_{compensated}} \approx 0.5T_{s_{uncompensated}}$

Compensated System: It was decided to use a proportional derivative controller.

$$G_p(s) = \frac{k}{.00002921s^3 + .0315s^2 + .5048s} = \frac{k'}{s(s+1062)(s+16.3)}, \zeta = \frac{-\ln(4.5/100)}{\sqrt{[\pi^2 + \ ln^2(4.5/100)]}} = 0.7025$$

Normalization of the coefficient of the s^3 term gives the relationship $k = 0.00002921k'$. This is because Matlab normalizes the leading coefficient in the denominator during factorization.

At ζ = 0.7025, the dominant poles are $s_{1,2=} -8.07 \pm j8.15 = -\zeta\omega_n \pm j\omega_d$

$$\zeta\omega_n = 8.07, \ \omega_{n_{uncompensated}} = \frac{8.07}{0.7025} = 11.788\, rad/s$$

$$T_{s_{uncompensated}} = \frac{4}{\zeta\omega_n} = \frac{4}{8.07} = 0.4957 sec, \ T_{s_{compensated}} = \frac{T_{s_{uncompensated}}}{2} = 0.2478\, sec$$

$$\omega_{n_{compensated}} = 2\omega_{n_{uncompensated}} = 24.3 \ rad/s$$

Figure 9 shows a point (indicated by the diamond) selected in the "acceptable zone" for the design. The "acceptable zone" is the unshaded area bounded by the constant damping ratio lines and outside the natural frequency circle. The design point was arbitrarily chosen to see how well the system will respond. A PD controller, $G_c(s) = s + \ Z_c$ is selected. At the design point, the

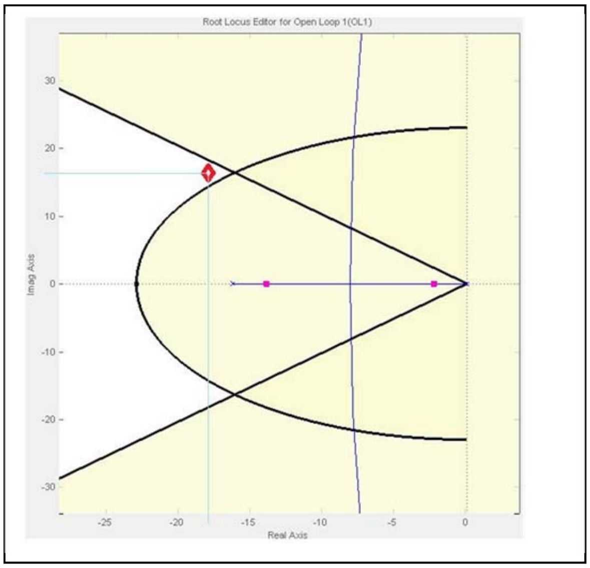


Figure 9. Dominant Poles of Uncompensated System and Desired Design Point.

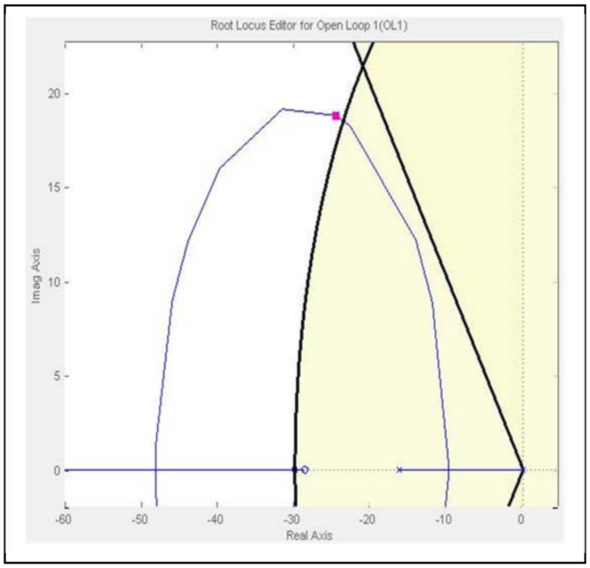
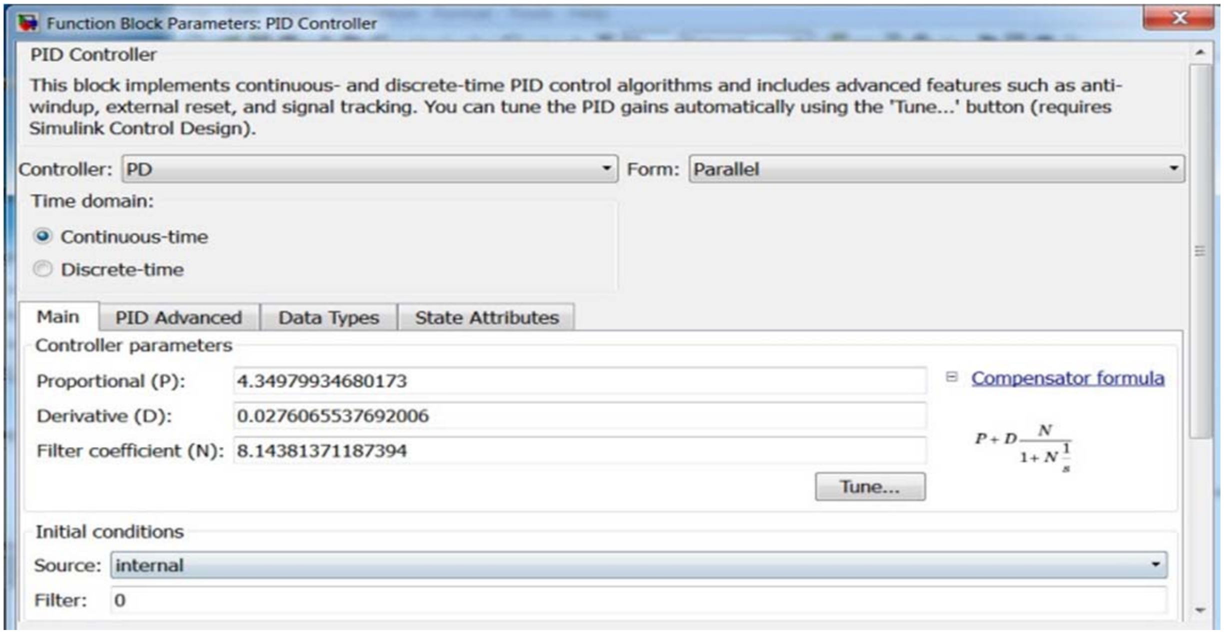


Figure 10. Compensated System with zero at -28.6


Function Block Parameters: PID Controller

PID Controller

This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).

Controller: PD

Form: Parallel

Time domain:

● Continuous-time

○ Discrete-time

Controller parameters

Proportional (P): 4.34979934680173

Derivative (D): 0.0276065537692006

Filter coefficient (N): 8.14381371187394

$P+D\frac{N}{1+N\frac{1}{s}}$

Initial conditions

Source: internal

Filter: 0

Figure 11. Parameters obtained using PID tuner with Filtering for Black line Follower

poles are $s_{1,2} = -18 \pm 16j$ (Figure 9), and the location of the zero was found from hand computations. Applying the angle criterion, the angle that the zero made with the real-axis was $\theta_{Z_c} = 56.5^o$, which corresponds to $Z_c = 28.6$ (Figure 10.) Unfortunately the calculated values for the gains did not quite satisfy the specifications. However the values were used as a starting point in the PID tuner. Figure 11shows the values generated for the PD controller. The NXT model of the compensated system is shown in Figure 12. In order to generate the uncompensated system the proportional controller gain was set to unity. A better approximation of the theoretical results may have resulted if the design point was the point of intersection of the damping ratio line and the natural frequency circle. The PID tuner values did satisfy the desired specifications. Figure 13 shows the voltage plot as the black line is traversed.

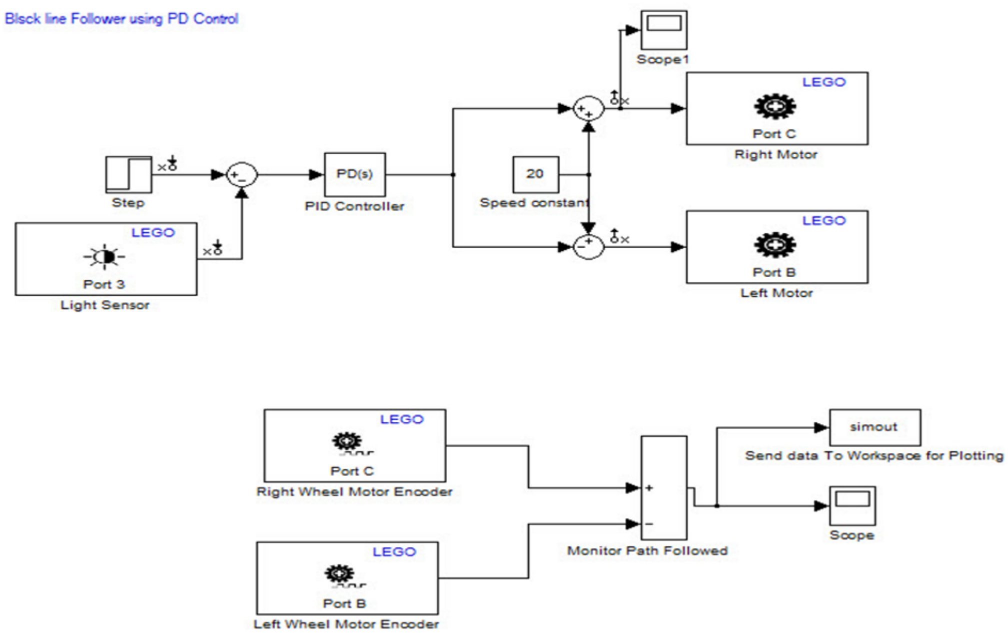


Figure 12. The Black-line Follower LEGO NXT Model with tuned PD Parameters

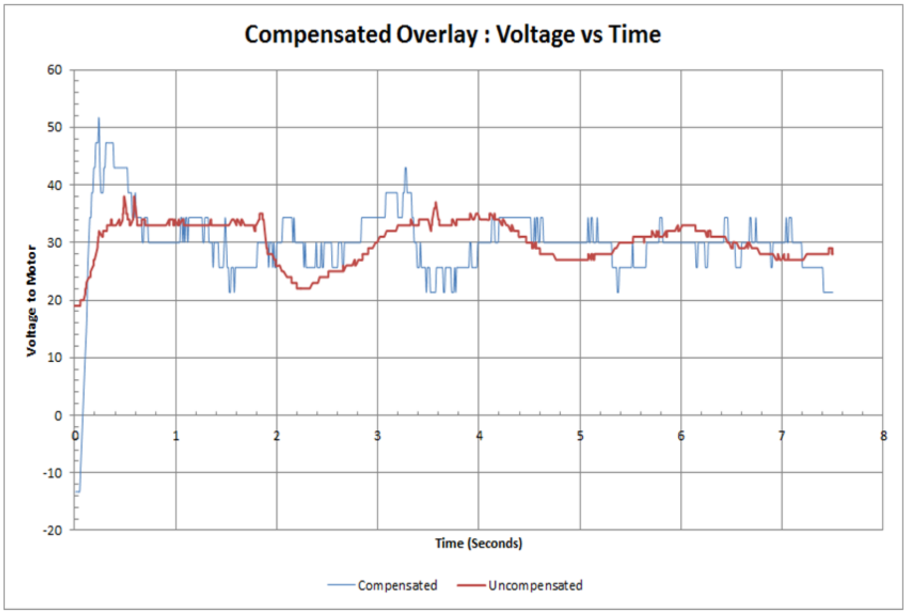


Figure 13. Voltage versus Time

Table 3. Uncompensated and Compensated Characteristics of the Black Line Follower

Parameter	Uncompensated	Compensated (Before Tuning)
Plant & Compensator	$\frac{k}{2.921(10^{-5})s^3 + .0315\,s^2 + .5048s}$	$\frac{k(s+28.67)}{2.921(10^{-5})s^3 + .0315\,s^2 + .5048s}$
Dominant poles	-8.07±j8.14	-18.3±j16
k	4.082	0.6147
$e(\infty)$	0	0
ζ	.7025	.753
ω_n	11.53	24.3
% OS	4.5	4.5
T_s (sec)	.4938	.2478
T_p (sec)	.3827	.1921
Zero	None	-28.67

<u>Summary:</u> In the uncompensated system the robot constantly veered off the black line. With proportional derivative control a smooth response was obtained as the robot followed the curved path. Design specifications were satisfied.

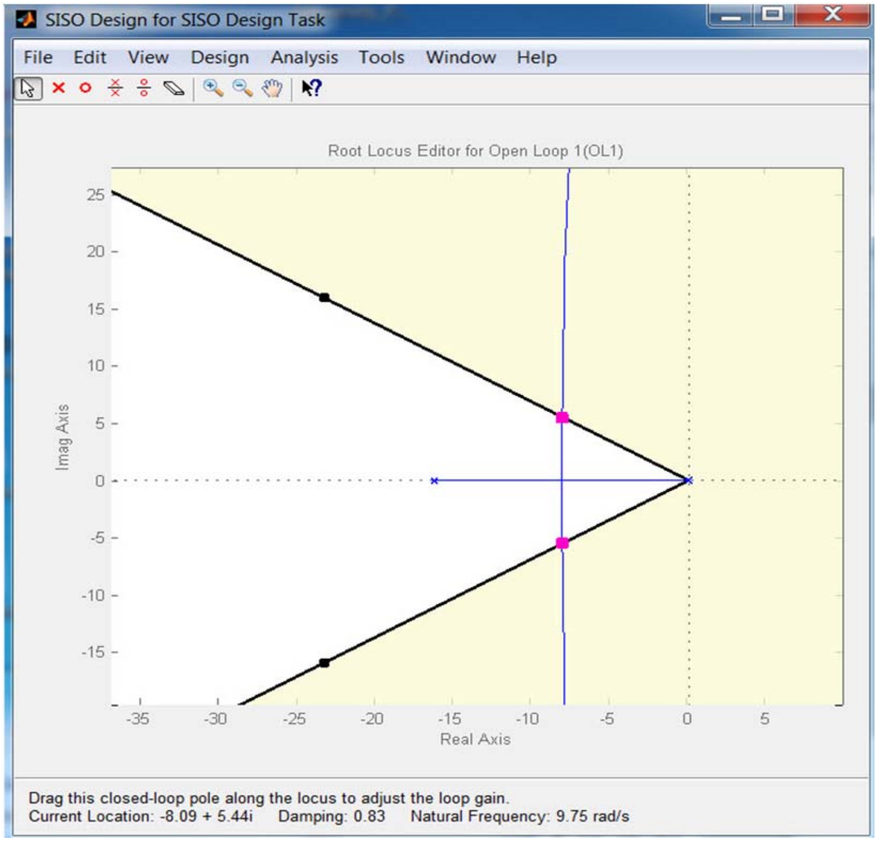
Figure 14. Robot accurately following a black line

Project 3: Pick and Place a Ball using Proportional Control

Objective**:** To perform the classic pick-and-place task. Specifically, the robot will rotate 90 degrees counter-clockwise about a central motor axis. A second motor will extend an arm that holds a third motor. The third motor will control an assembly that grabs the ball. The second motor will then retract the arm. The first motor will rotate the system 90 degrees clockwise to the starting location and, with the use of the second and third motors, place the ball on a holding rack.

Constraints to be satisfied: When the system is uncompensated, the movement of the arm is sluggish and does not reach or retract to the desired positions with precision. The response of the motors to the signal sent by the NXT will be improved by an appropriate controller so that the system's performance is optimized. In order to make the system more stable and predictable it was necessary to guarantee an overshoot of less than 1% for motor 1 and motor 2.

Compensated System: A proportional controller was used to see if the design requirement of 1% overshoot will be satisfied. At the design point the dominant poles are $s_{1,2} = -8.09 \pm j5.44$ (Figure 15) with a proportional gain *k* = 2. Figure 16 shows the step response plot.



Drag this closed-loop pole along the locus to adjust the loop gain.
Current Location: -8.09 + 5.44i Damping: 0.83 Natural Frequency: 9.75 rad/s

Figure 15: Root locus showing design point for the compensated system.

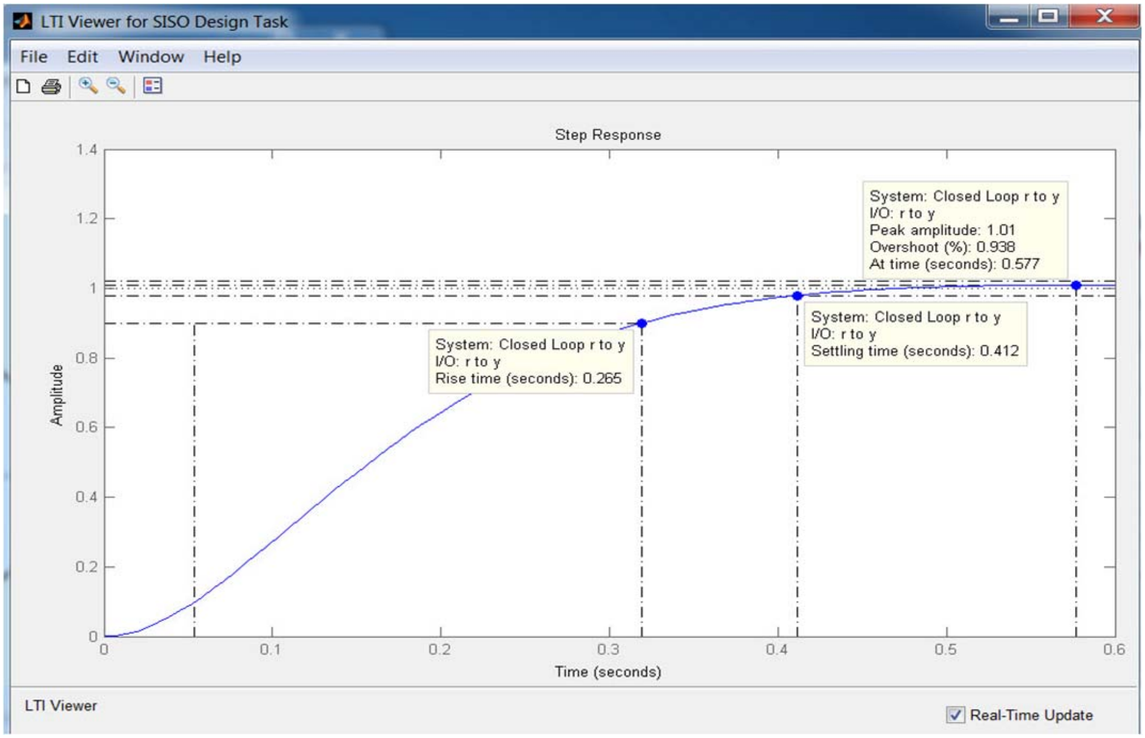


Figure 16: Time Response Plot for the Compensated System

Table 4. Compensated System Characteristics of Classic Pick and Place

Parameter	**Compensated (Motor 1 & 2)**
Plant & Compensator	$\frac{k}{2.921(10^{-5})s^3 + .0315\, s^2 + .5048s}$
Dom, poles	-8.09 ± j5.44
k	2.95
ζ	.8261
ω_n	9.793
% OS	0.938
T_s (sec)	0.412
T_p (sec)	0.577
T_r(sec)	0.265

The time sequence for the pick and place process is controlled in the respective step input blocks, and designated by numbers 1, 2, …, 8 (Figure 17.) This model was generated with Matlab version R2013b and a slight modification of the gain to 2.49 was needed for motor 2. In Matlab version R2012a the gain for each motor was set to 2.95.

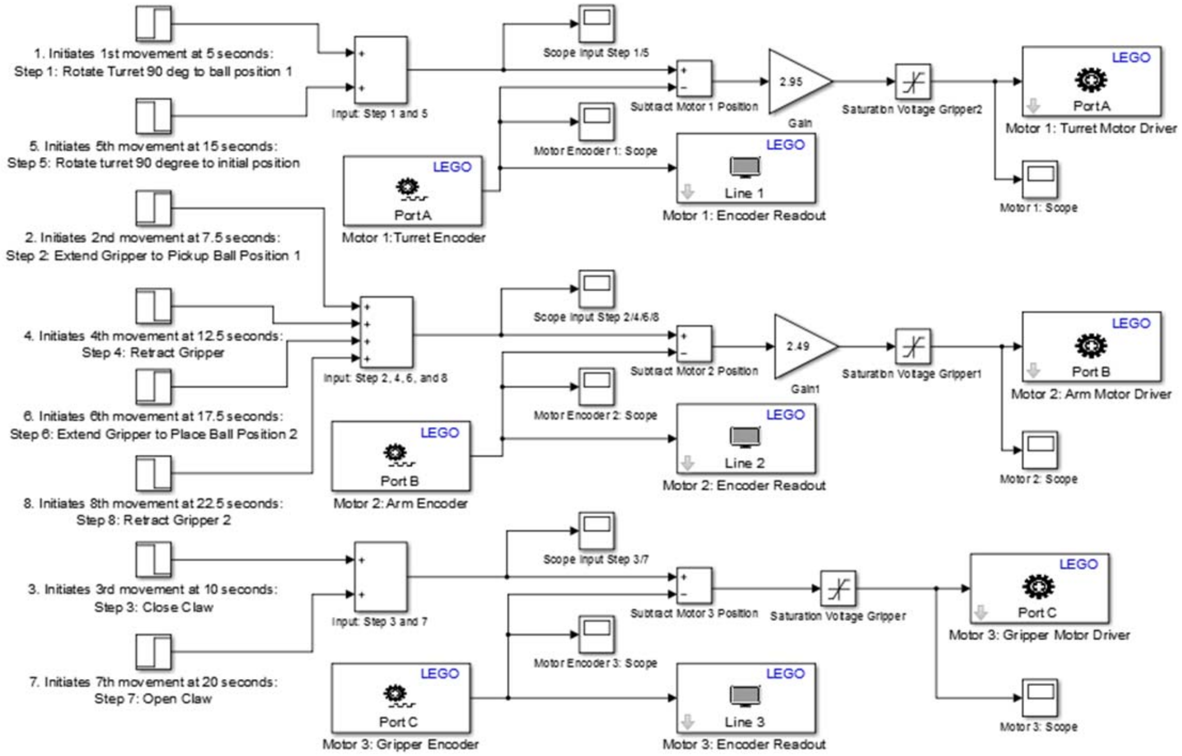


Figure 17. Compensated LEGO Mindstorms Model - Pick & Place

<u>Summary:</u> The proportional controller implemented to carry out the Pick-and-Place task was sufficient for satisfying the system specifications. The error in the reaching/retracting arm was eliminated as evidenced by the arm's return to its initial position after delivering the ball. The reduction in percent overshoot resulted in the gripper no longer knocking the ball off the holder nor misplacing the ball at the final destination.

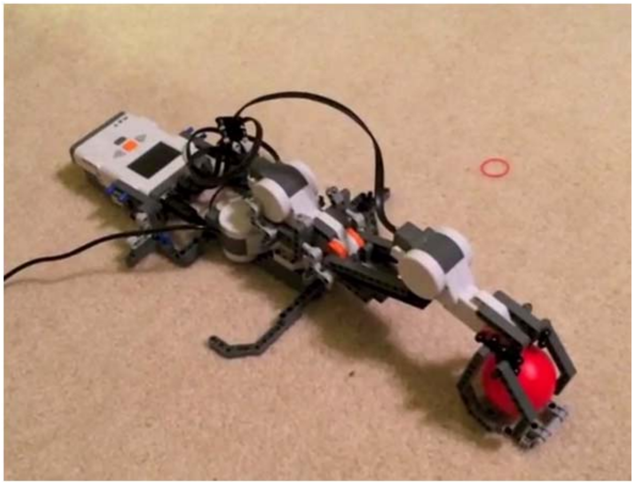


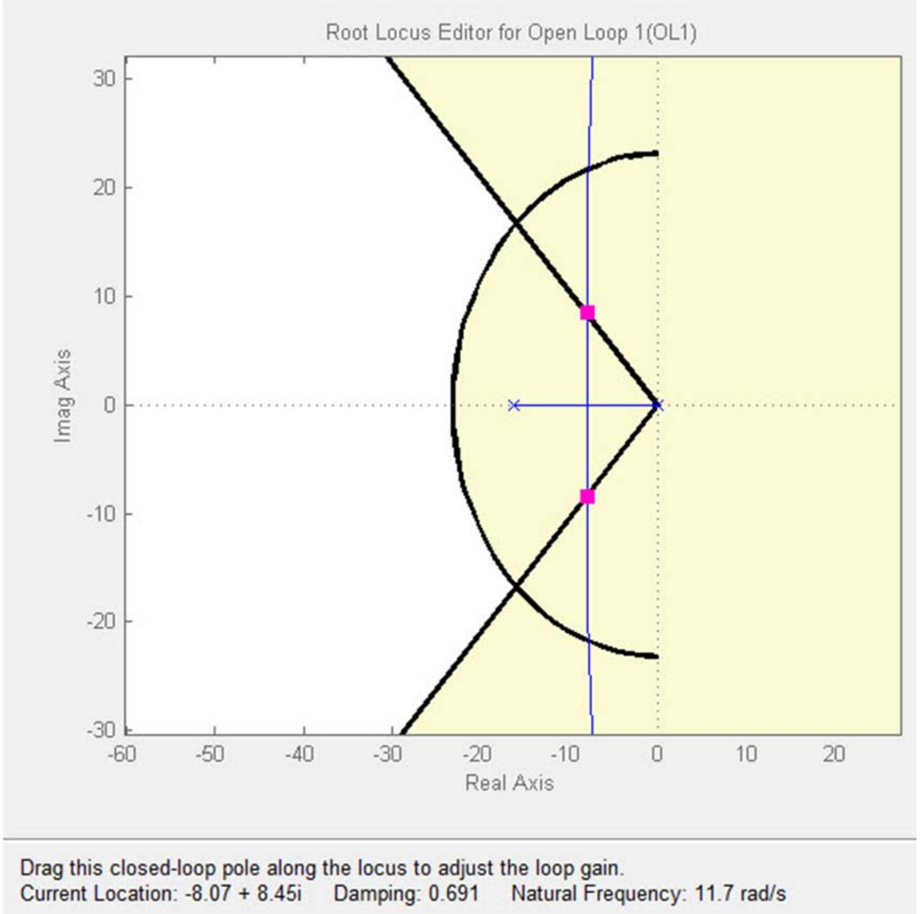
Figure 18. Compensated System. Ball is picked up from the red circle and deposited in the holder.

Project 4: Color Detection, Grab and Drop Red Ball via PD Control

Objective**:** To design a robotic claw that is capable of rotation in a horizontal plane, determination of the color of pre-positioned red and blue balls, selection of the red ball, and securing and delivering the red ball to a final storage location. The claw will use two motors, one to control its angular rotation, and the other to operate the claw mechanism. The robot will use the light sensor to detect the color of a ball. When the red ball is detected, the ball is retrieved and returned to a final location that is midway between the ball mounts.

Constraints to be satisfied: The first observation was that the uncompensated system had an extremely slow response. The system is overdamped , the rise time and settling time are very high, and these high values equate to a large portion of time that the system is not performing at its maximum potential. Since the arm only takes about a second to get to the first ball, an approximate value of 0.25 seconds settling time, T_s, deemed appropriate. Also for smooth transitions a 5 percent overshoot was chosen.

Uncompensated System: The root locus for the uncompensated system shows that at 5 percent overshoot the dominant poles are at $s_{1,2} = -8.07 \pm 8.454j$ with a gain, $k = 4.2377$.



Drag this closed-loop pole along the locus to adjust the loop gain.
Current Location: -8.07 + 8.45i Damping: 0.691 Natural Frequency: 11.7 rad/s

Figure 19. Root Locus of Uncompensated System with dominant poles at 5% Overshoot.

Compensated System: Proportional control was first attempted to see if the specifications will be satisfied. Figure 20 shows two unit step responses. For $k = 5$, $T_s = 0.473$ sec, overshoot = 7.57%, and for $k = 100$, $T_s = 0.571$ sec, overshoot = 68.9%, demonstrating that a proportional controller will not satisfy the specifications.

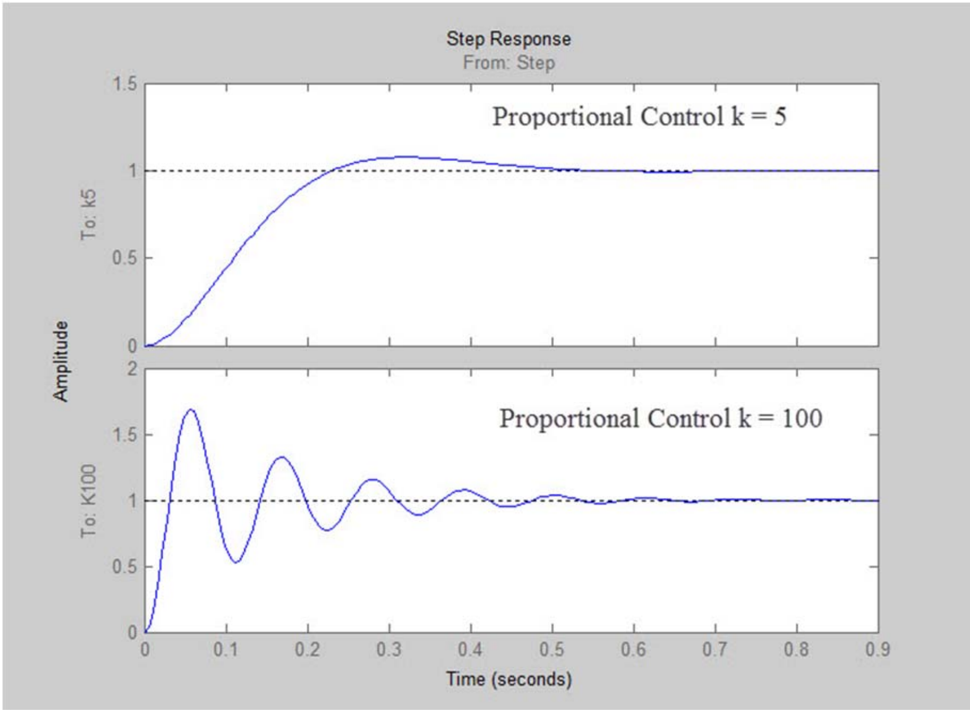


Figure 20. System Responses with Proportional Control

While k was large (>10) the system had a very large percent overshoot, and although the rise time was very fast the settling time was greater than the desired 0.5 seconds. As k decreased, the overshoot decreased, but the rise time increased and the settling time remained unchanged. This eliminated the option of using a P controller, and since a PI had already been ruled out due to the steady-state error being zero, a PD controller was chosen to satisfy the constraints. Calculations to determine values for the PD controller were conducted.

PD Controller Computations:

$$\%OS = 5\%, \qquad T_s < 0.25\,sec, \qquad \zeta = \frac{-\ln(5/100)}{\sqrt{[\pi^2 + ln^2(5/100)]}} = 0.6905$$

$$T_s = \frac{4}{\zeta\omega_n} < 0.25sec, \qquad \omega_n > \frac{4}{0.25\zeta} = 23.17\,rad/sec$$

For the design, $\omega_n = 23.2\,rad/sec$ was chosen.

The *sisotool* plots (Figure 21) show a point selected in the acceptable zone of the root locus for the design. The calculation of parameters of a PD controller, $G_c(s) = s + Z_c$ for the selected design point, $s_{1,2} = -16.0196 \pm 16.78j$ gave $\theta_{Z_c} = 43.74^o$ and $Z_c = 33.55$. The *sisotool* compensator editor is shown in Figure 22.

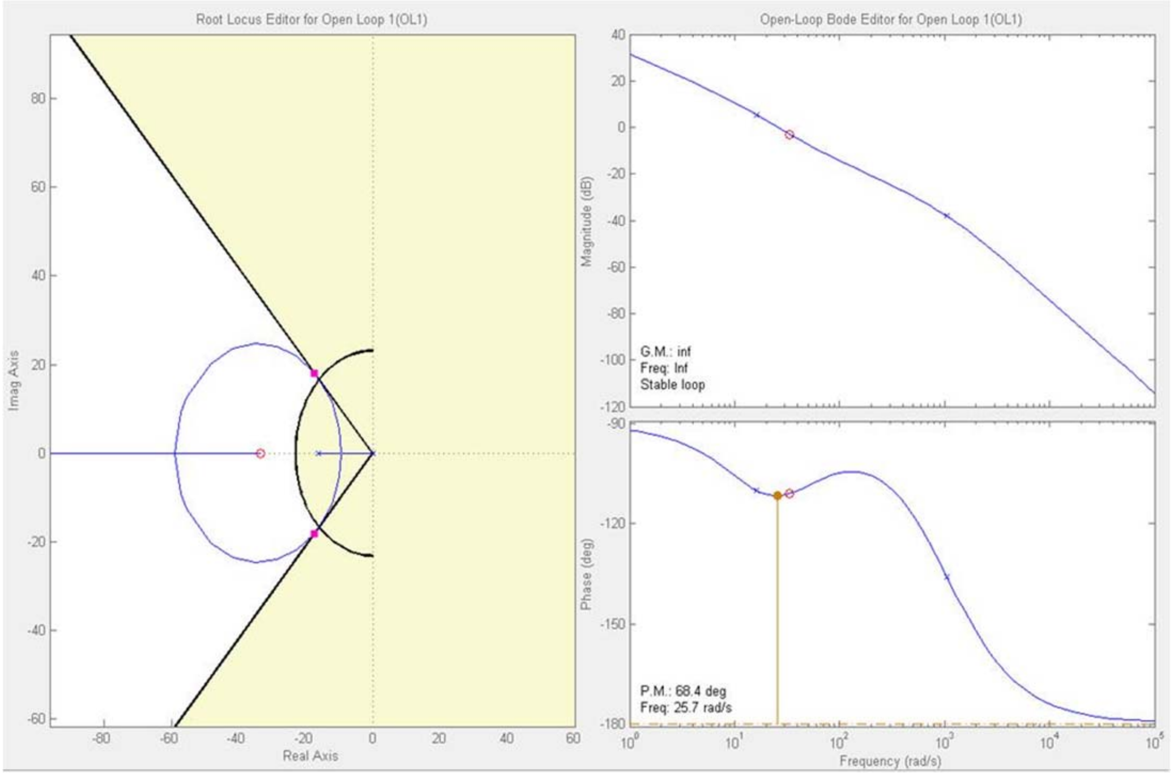
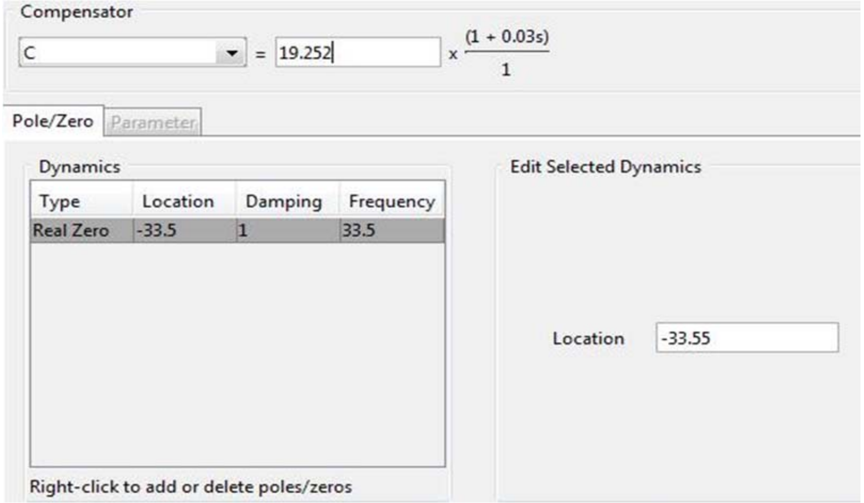


Figure 21.Root Locus and Bode Plot using SISOTOOL



C = 19.252 × $\frac{(1 + 0.03s)}{1}$

Dynamics

Type	Location	Damping	Frequency
Real Zero	-33.5	1	33.5

Location -33.55

Figure 22. Compensator Editor showing Gain and Zero Location

Using the PID auto tuning in Simulink gave the final PD controller values (Figure 23) that were used in the real-time design. The hand-calculated values were comparable to those computed by the tuner. Figure 24 shows the time response plots for the uncompensated and compensated systems.

Controller parameters

Proportional (P): 20.37

Derivative (D): 0.8056

Filter coefficient (N): 70.42

Compensator formula

$$P + D\frac{N}{1+N\frac{1}{s}}$$

Figure 23. PID Tuner Compensator Values

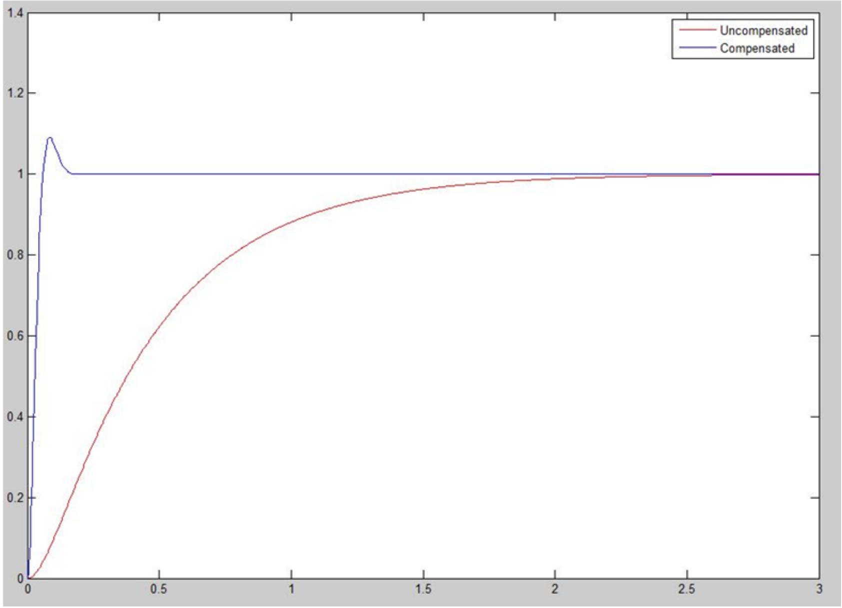


Figure 24. Time Response of Uncompensated and Compensated Systems

The predicted characteristics are shown in Table 5.

Table 5. Predicted Characteristics – Color Detect, Grab and Drop Red Ball

Parameter	**Uncompensated**	**Compensated (before tuning)**
Plant & Compensator	$\frac{k}{s(s+16.27)(s+1062)}$	$\frac{k(s+33.55)}{s(s+16.27)(s+1062)}$
Dominant Poles	-8.07 ± 8.454j	-16.02 ± 16.78j
k	4.2377	19.252
k_p	∞	∞
e(∞)	0	0
ζ	0.6905	0.6905
ω_n (rad/s)	11.69	23.20
% OS	5%	5%
Ts (sec)	0.4955	0.2497
Tp (sec)	0.3715	0.1872
Third Pole	-1,062	-1,043
Zero	None	-33.55
Comments	Second-order approx. OK	Second-order approx. OK

The NXT model shown in Figure 25 is for the compensated system. To create the uncompensated system a proportional controller with gain, $k = 1$ was used.

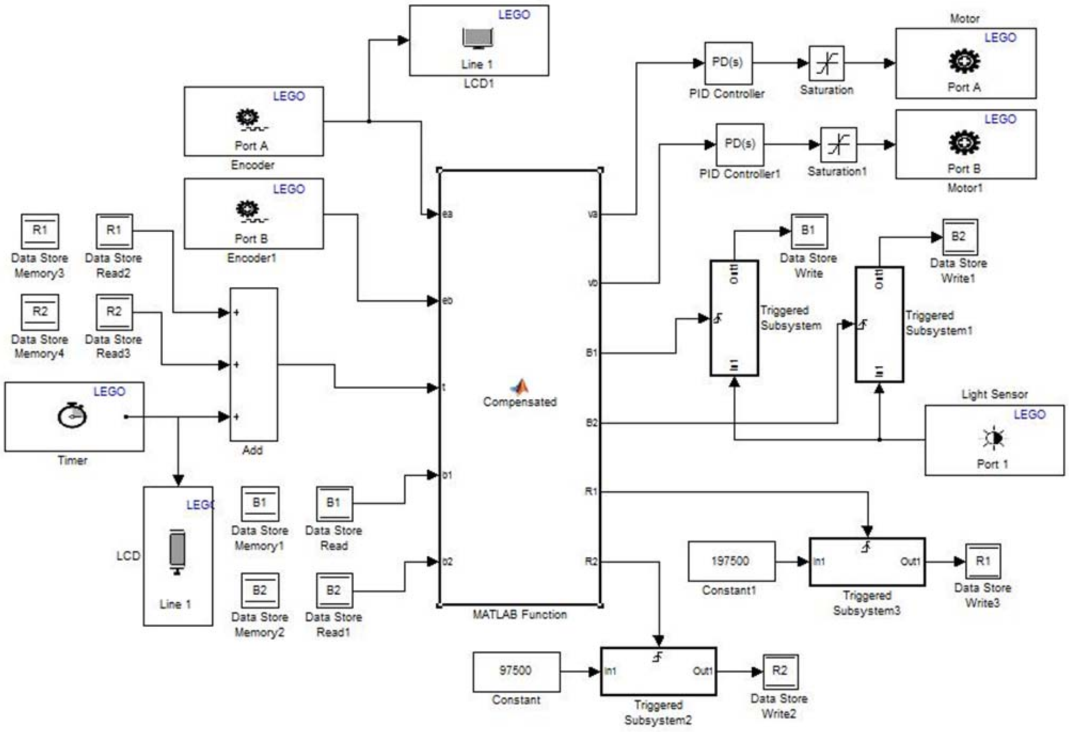


Figure 25. LEGO NXT Model for the Compensated System

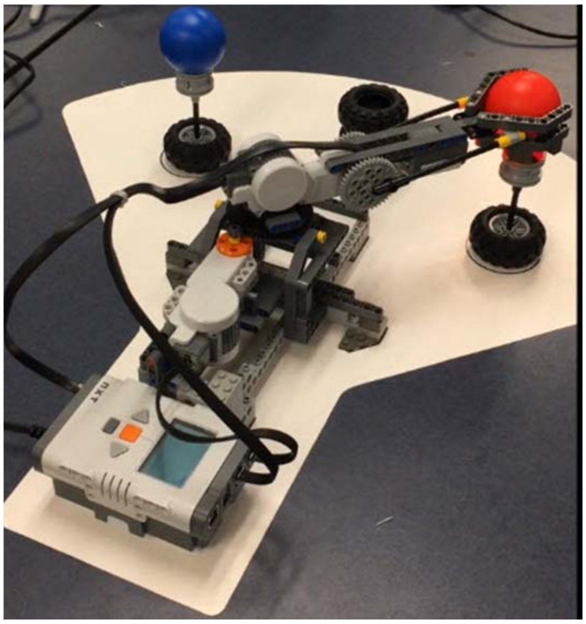


Figure 26. Compensated System. Red Ball is detected and ready for delivery on the tires located midway between the ball mounts.

<u>Summary:</u> By using proportional control the reaching/retracting arm performed smoothly and returned to its initial position after delivering the red ball on the tires mounted midway between the balls. The percent overshoot was decreased such that the robot no longer knocked the ball off the holder or misplaced the ball at the delivery point. Regardless of whether the red ball is placed on the left-side or right-side mount, the robot successfully detected, picked up and delivered the red ball to the final destination.

Controller Parameter Selection and Trial-and-Error Minimization

The choice of parameters is learned through the project. In the first phase of the project the behavior of the uncompensated system is observed. The steady-state error is negligible therefore the emphasis is generally on reducing percent overshoot, and improving rise time or settling time. In the second phase, the *Characteristics of Types of Cascaded Compensators* table, Nise[6], is used to select the appropriate types of compensators that will improve the transient behavior and satisfy the constraints. The students are required to perform theoretical analysis starting with the simplest type of controller, the proportional controller. Hand-computations via the root locus technique are conducted to see if a proportional controller will suffice. Simulations using Matlab's *rltool* or *sisotool* are also conducted. If a proportional controller is not adequate, further root-locus analyses using other appropriate compensators are conducted. To isolate the area on the root locus that will best satisfy the constraints, the constant damping ratio lines and natural frequency circle are used. Specifically the acceptable zone is the area outside of the natural frequency circle and to the left of the constant damping ratio lines. Figures 4, 10, 15, and 21 show the "acceptable zones" in the respective root-loci from which compensator parameters are selected. The red dot indicates the point chosen for a particular project. Knowing that truncation or round-off errors are ubiquitous in hand-computations and computer simulations, good performance may not be achieved when the values for the compensator parameters are used in the control of the real system. In such cases, the parameter values that are obtained from the "acceptable zone" are used as starting points in the Simulink PID controller block. The starting points provide a good set of initial values for the PID tuner, which means that only minor tuning is needed to obtain good values for the real system. If the acquisition of the starting values using theoretical analysis is bypassed, there is a tendency for students to resort to trial and error. Two main disadvantages of the trial and error method are (i) it is time consuming and (ii) the parameter values obtained may be far removed from what is needed to meet the given requirements. In summary, obtaining starting values from theoretical calculations provides parameters that are very close to those needed to satisfy the constraints of the real system, and trial and error is minimized.

Assessment

This assessment is based on the aforementioned learning outcomes. The weights for the three main categories used for the evaluation of the student's understanding of the material are three tests (60%), short quizzes using clickers (10%), and a project (30%.) Emphasis is placed on the student's ability to control a dynamic system and the main areas considered in the evaluation of the project are the powerpoint presentation, the robot demonstration, and the technical report. The instructor critiques the presentations and other class members are encouraged to ask questions of the presenting group. Students are required to implement feedback received during the presentation in their technical report which is due a few days after the presentation. Figures 27 and 28 show the impact of the project on student learning for two out of the three semesters that the NXT project has been used for real-time control system design. In both semesters approximately 88% of the students had a satisfactory, good, or very good grasp of controller design at the completion of the course. This is an average increase of 9% over previous semesters in which a final exam was administered in lieu of the NXT project.

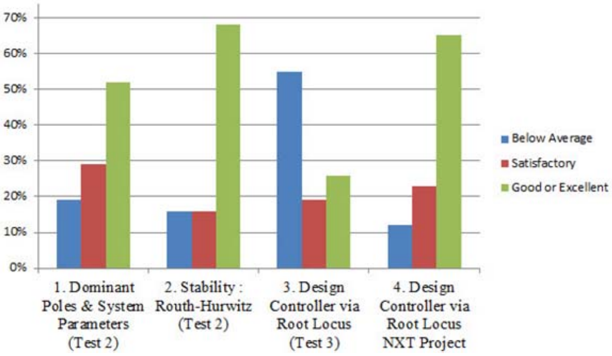


Figure 27. Student Performance – Semester 1

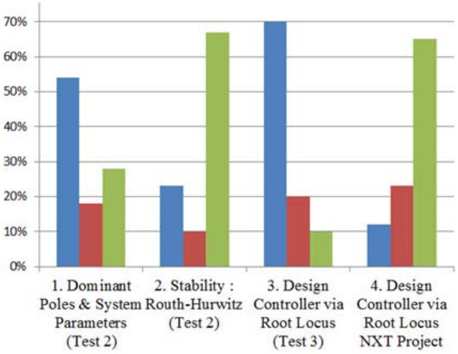


Figure 28. Student Performance – Semester 2

Students commented that completing the project provided them with the opportunity to more thoroughly review and apply the principles. Some students recommended that a laboratory component be added to the course. However instructors teaching other sections of the course do not have access to the LEGO Mindstorms, and budget constraints prevent the acquirement of more kits. One solution might be for students to buy the NXT microprocessor, known as the brick, and for the department to supply the other LEGO pieces. Another suggestion was that the NXT be more fully integrated into the course by starting simple projects in the third week when system parameters are discussed in the lectures; plans are underway to do this beginning in the spring semester 2015. Table 6 shows the table of contents for technical writing that is distributed to the students and used as the project grading rubric.

Conclusions

The LEGO NXT project has been effective in helping students to better understand and apply the principles of automatic controls. Observing students (i) engage in group discussions as they attempt to convince others of the best approach to take in the design process, (ii) make a conscientious effort to develop projects that would be superior to those of their peers, and (iii) make more use of office hours to discuss their progress, has provided the impetus toward developing an elective course in advanced automatic controls that the students are requesting. The inclusion of the project in this course has given credence to the adage “Tell me and I forget, teach me and I may remember, involve me and I learn.” Future challenges include finding ways to improve the student’s performance in the written tests, and apply other techniques to help implement system identification in the design process. Some faculty members have already suggested that collaboration be established among faculty who teach the automatic controls course and the department is currently in the process of acquiring a few of the newer LEGO EV3 kits. It is anticipated that such collaboration will help address the aforementioned challenges.

Table 6. Technical Report Grading Rubric

Title Page
Table of Contents
Table of Figures
Objective

1. Uncompensated System – Theoretical Analysis
   a. Block Diagram (Simulink Model)
   b. Root Locus Sketch and Hand-Computations
   c. Uncompensated Root Locus via *rltool* or *sisotool*
   d. Time-Response Plot
2. Uncompensated System and Design Requirements for the Compensated System.
3. Compensated System – Theoretical Analysis
   a. Hand computations for Compensator
   b. Block Diagram (Simulink Model)
   c. Root Locus for Compensated System via *rltool* or *sisotool* (Final Design)
   d. Time Response Plots (Compensated & Uncompensated responses on one plot)
4. Summary of Predicted Characteristics: Uncompensated and Compensated Systems.
5. Compensated System – LEGO MINDSTORMS Model
   a. Block Diagram (Simulink LEGO MINDSTORMS NXT Model)
   b. Detailed Steps needed to Build and Execute the Model
   c. Time Response Plot via Matlab
6. Conclusions
7. Recommendations
8. References
9. A Short Video of the Uncompensated and Compensated Systems.

Acknowledgements

Project examples presented in this paper were contributed by the following undergraduate students: Kenny Bruner, Christopher Bump, Brian Cormier, Ian Cullity, William Foster, Ian Heim, Jeremy Kearney, Truman Lum, Cristina Lupercio, Simrata Randhawa, Eric Fornalski, Jeremy Steinmiller Brian Tracy, and Jeremiah Vargas.

Bibliography

[1] Gordon, Bernard. *MIT White Paper on UK Approaches to Engineering Project-Based Learning*. Apr. 2010. Web. 7 Mar. 2015.

[2] Fernández-Samacá[II]; Liliana, José Miguel Ramírez[I], and Martha L. Orozco-Gutiérrez[I]. *Project-based learning approach for control system courses*. Feb.2012. Web. 7 Mar. 2015

[3] Enikov, Enikov,Vasco Polyzoev, and Joshua Gill. *Low-Cost Take-Home Experiment on Classical Control Using Matlab/Simulink Real-Time Windows Target*. 2010. Web. 7 Mar. 2015.

[4] Golnaraghi, Farid, and Benjamin Kuo. Automatic Control Systems. New Jersey: John Wiley & Sons, Inc., 2010. Print

[5] n.p. *Robotics with the Boe-Bot: Student Guide*. Parallax.com. 2010. Web. 16 Jan. 2015.

[6] Nise, Norman. *Control Systems Engineering*. New Jersey: John Wiley & Sons, Inc., 2011. Print.

[7] MathWorks Simulink Team. *Simulink Support Package for LEGO MINDSTORMS NXT hardware*. 18 Sep. 2014. Web. 16 Jan. 2015.

[8] Mechatronics Videos. *Simulink Modeling of a LEGO DC Motor*. Online video clip. *Youtube*. Youtube, 1 Apr. 2013. Web. 16 Jan. 2015. https://www.youtube.com/watch?v=-izDVSCWhXI

[9] n.p. *Get Started with LEGO Mindstorms and Simulink*. Makerzone: Matlab & Simulink Project Resources. n.d. Web. 16 Jan. 2015.

[10] Cheever, Eric. *Linear Physical System Analysis*. Swarthmore College. n.d. Web. 16 Jan. 2015.

[11] Kearney, Jeremy. CSUS Control System Design (Lego + Simulink). Online video clip. *Youtube*. Youtube, 18 Aug. 2014. Web. 3 Apr. 2015. https://www.youtube.com/watch?v=I49FYUFkD4E

[12] Eke, Estelle. CSUS Control System Design (LEGO + Simulink). Online video clip. *Youtube*. Youtube, 2 Apr. 2015. Web. 3 Apr. 2015. https://www.youtube.com/watch?v=4X-Su1UFaqE

[13] Cullity, Ian. LEGO NXT - Online video clip. *Youtube*. Youtube, 12 Dec. 2014. Web. 3 Apr. 2015. https://www.youtube.com/watch?v=Z_rJe2zGUWg&feature=em-upload_owner

[14] itzBump. Lego Mindstorms NXT - CSUS Control Systems Design Final Project. Online video clip. *Youtube*. Youtube, 30 Mar. 2015. Web. 3 Apr. 2015. https://www.youtube.com/watch?v=IKfm_O0wXQ4