A LEGO based Undergraduate Control Systems Laboratory

S.R.Manikandasriram, *Student Member, IEEE,* Bharath Bhikkaji, *Member, IEEE,*

***Abstract*—The establishment of a laboratory for an introductory course on Control Systems incurs huge costs in terms of expensive hardware, large real estate and costly maintenance. Due to limited affordability, institutes are forced to convert a hands-on laboratory course into a demonstration class. This paper discusses the feasibility of using the LEGO Mindstorms NXT kit as a platform for introducing control theory concepts to engineering students. A set of experiments for explaining introductory topics in Control Engineering like System Identification, Frequency Analysis, PID Control and Bode Plots have been discussed. The paper also discusses the use of NXT kits for testing advanced controllers and for building low-cost alternatives for classic control problem setups like Inverted Pendulum and Pendulum on a Cart systems.**

***Index Terms*—Control System experiments, Control System education, LEGO NXT, ROBOTC, System Identification, Classic Control Theory, Frequency Analysis**

I. Introduction

THIS paper discusses the feasibility of using LEGO Mindstorms NXT kits as a platform for Control Engineering. The set of experiments described here demonstrate the versatility of the LEGO NXT platform, which can be used to implement most of the control theory concepts involved in an undergraduate and graduate course on Control Engineering. There are various software options for working with the LEGO NXT Kit.

II. ROBOTC PID Controller

In this experiment, the students observe the functioning of the closed loop PID Controller available in ROBOTC which is used for speed control of the *Interactive Servo motor*. The feedback in this system is provided by the *rotary encoder* present in the *Interactive Servo motor* which gives 360 counts per single revolution of the motor. This information is used by the "*NXT Intelligent Brick*" to calculate the angular position of the motor (with 1 degree accuracy) and also the instantaneous speed of the motor. The formula used for speed calculation is

$$speed = \frac{\Delta\theta}{\Delta t} \text{degrees/sec} \tag{1}$$

To observe the functioning of this inbuilt PID controller, a minimal setup as shown in Figure 1 on page 1 is required.

The LEGO motors are assigned a power rating between -100 and 100 with a negative value denoting reverse direction. Under ideal conditions, the maximum speed of the LEGO motors is 1000 degrees per second when the power rating is set at 100%. The inbuilt PID controller provides consistent speed by continuously adjusting the raw power sent to the motor.

Fig. 1. Experimental setup for observing in-built PID Controller

To enable PID Speed Control and set the maximum regulated speed, the following commands can be used

```
nMotorPIDSpeedCtrl[motorB] = mtrSpeedReg;
nMaxRegulatedSpeedNxt = 750;
```

It is recommended to keep this value below 1000 since the battery voltage may not remain at 100%.

The LEGO NXT has a *Datalog* feature which can be used to log sensor data and store status information which can later be exported to a PC for post-processing. In this experiment, the encoder counts and raw power sent to the motor are logged every Δtms. A sample ROBOTC for the same is provided below.

```
task main()
{
  nMaxRegulatedSpeedNxt = 500;
  // Reset the Motor Encoder
  nMotorEncoder[motorB] = 0;
  nMotorPIDSpeedCtrl[motorB] = mtrSpeedReg;

  int motorRAWpower, motorDegrees;
  motor[motorB] = 50;    // power level of 50/100
  time1[T1] = 0;         // Timer

  // Allocate memory for datalog
  nDatalogSize = 1600;

  while (time1[T1] < 10000) {
```

```
    motorDegrees = nMotorEncoder[motorB];
    motorRAWpower = motorPWMLevel[motorB];
    // store value to Datalog
    AddToDatalog(1,motorPIDdegrees);
    AddToDatalog(2,motorPIDpower);
    wait1Msec(50);      // log data every 50 ms
  }
  motor[motorB] = 0;    // Stop the motors
  SaveNxtDatalog();
}
```

The output of the program is a data log *DATAnnnn.rdt* which can be exported to a PC using the *File Management Utility* available in the ROBOTC Development Environment under *Robot→NXT Brick→File Management Utility*. The *Spreadsheet Upload* button converts the *.rdt* file into a *.csv* file which can be processed using any Spreadsheet Processors. The data is stored as *Key,Value* pairs which constitute the two columns in the exported CSV file. The datalog sequentially stores data with every call to the

```
AddToDatalog(<index>,<data>);
```

creating a new entry in the file with a *key* which is proportional to the *index* and the *data* stored as a $16-$bit unsigned integer.

III. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] http://education.lego.com/en-us/preschool-and-school/secondary/11plus-mindstorms-education/rcx
[2] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.