

A Hands-on Laboratory for Introductory Automatic Control Courses

Chyi-Shyong Lee*, Juing-Huei Su*, Cheng-Chang Hsieh*, Kuo-En Lin*, Jia-Hao Chang*, and Gu-Hong Lin*

**Lunghwa University of Science and Technology, Taoyuan county, 333
Taiwan (Tel: 886-8209-3211 ext. 5400; e-mail: suhu@mail.lhu.edu.tw).*

Abstract: This paper presents a hands-on laboratory for introductory automatic control courses, which will be offered to the third or fourth year students of the department of electronic engineering in the Lunghwa University of Science and Technology. The aim of the hands-on laboratory is to help reinforce the learning process of the students in introductory automatic control courses from a practical point of view. Therefore, a low cost (~\$USD 60) but versatile educational platform based on the dsPIC microcontroller is devised as a learning aid to implement practical control algorithms for motion control systems. A monitor and control software running on the PC side is also developed to collect experimental results from and send commands via universal serial bus (USB) to the platform. All the necessary softwares are given at no cost to the students during the class. In addition to the theoretical part of automatic control systems and microcontroller structures, the hands-on laboratory consists of four parts of exercises which are used to help students learn step by step the automatic control theory, how to implement control algorithms in a microcontroller, and the performance requirements of motion control systems.

1. INTRODUCTION

Control system theories and designs are often based on abstract mathematical concepts and physics. They are used to help students get familiar with the dynamics of the system to be controlled, and then find ways to design a successful feedback controller to meet prescribed specifications. Moreover, the feedback controller is often constructed nowadays via software algorithms in microcontrollers. This is because of the convenience of updating the algorithms in software forms and the fast performance improvements of microcontrollers. Designs of control systems therefore involve knowledge of mathematics for modeling, feedback controller design, and knowledge of microcontrollers for implementation of control algorithms. In order to make the education of automatic control systems both conceptual and experiential (Bernstein, 1999), fast development and educational tools for constructing a automatic control system to validate the theory in textbooks and to learn practical issues in implementation are necessary for students and engineers to quickly grasp the practical working principles of a successful automatic control system.

Although the Quanser Inc. (www.quanser.com) has recently developed the QET dc motor control trainer that demonstrate control fundamentals by using a wide range of control methods, the product is quite expensive (~USD\$ 4000) and not affordable to our students. Chen and Naughton (2000) also proposed an undergraduate laboratory platform for control system design, simulation, and implementation based on the popular MATLAB/SIMULINK environment. The platform presents an excellent environment which allows students to focus all their effort on learning the control system theory when controlling real world systems. The only drawback of this platform is that students can not have a clear picture of how to construct the controller in a micro-

controller and the corresponding programming and implementation issues. This is because the environment is built on top of the real-time operating system, RTLinux, the MATLAB/ SIMULINK software, and a DSP board to handle all the necessary signal input and output functions. Therefore, all the students have to do is to code the corresponding control algorithms as MATLAB script languages (.m files) or SIMULINK models. The environment is easy to use for students, but difficult for teachers to maintain. Moreover, it is impossible for those schools which do not purchase the MATLAB/SIMULINK software to adopt the approach as an educational tool. Tornambè (2005) uses the basic stamp II (Parallax, Inc.) as a simple tool to implement control laws for dc motors. The experimental setup in (Tornambè. 2005) is quite simple and inexpensive. Students can easily write codes by using the high level PBASIC programming language to control dc motors. The main drawback of this platform is that it does not support interrupts (Parallax, Inc., 2000), which is important to accurately control the timing for digital control algorithms. Therefore, the reaction time of each iteration of the digital control algorithm by using the approach (Tornambè. 2005) depends heavily on what instructions are executed.

In order to remedy the abovementioned problems and to benefit those undergraduate students who are in general not wealthy, this paper presents a hands-on laboratory for introductory automatic control courses which includes a low-cost educational platform to meet the needs. The project is also supported free of charge those necessary microcontrollers by the local branch of Microchip Technology Inc. The educational platform is a revision of the one proposed in (Lee, 2006) which is cheap and simple enough that every student can build his/her own platform. The hands-on laboratory is designed for students who have

preliminary knowledge about microcontrollers and programming skills. This is why the course will be offered to the third or fourth year students. The servo controls of dc-motors and the furuta pendulum (Pan, 2000) are used in the hands-on laboratory to lead students step by step to get familiar with the introductory automatic control theory and the corresponding practical implementation issues.

2. OVERVIEW OF THE EDUCATIONAL PLATFORM

2.1 Hardware

The devised educational platform for the hands on laboratory whose block diagram is shown in figure 1 is a single board computer and it costs about USD\$60 (see table 1 for cost analysis). It consists of a dsPIC microcontroller, a USB port to communicate with a PC, an in system programming (ISP) port for downloading programs from a PC to the 48K byte flash memory, 6 pulse width modulation (PWM) output channels to drive power switches, 9 A/D ports for accessing physical signals, a built in encoder input port, a optional LTC-2622 to provide dual 12-bit D/A channels, and several digital I/O ports. In this educational platform, the H-bridge driver (LMD18200) capable of accepting up to 55V voltage source and driving up to 3A loads is used as a DC motor driver for convenience, though it is about one-third of the cost of the educational platform. Another benefit of using the LMD18200 H-bridge driver for the students is that the dead time to prevent short circuit of H-bridge driver legs (N. Urasaki *et al*, 2007) is not necessary.

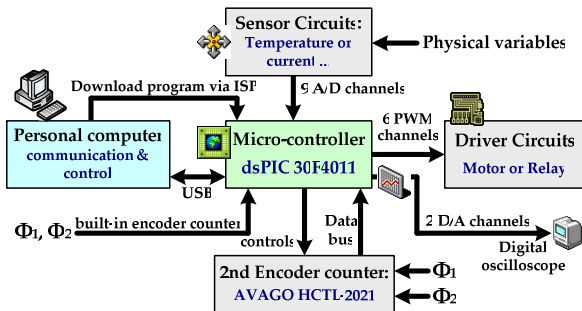


Fig. 1. The hardware blocks of the educational platform

Table 1. The cost analysis of the educational platform

Components and subsystems	USD\$
dsPIC30F4011	8
USB communication circuit	10
In system programming circuit	2
PCB board	5
LMD18200 H-bridge driver circuit	10
A dc motor with an optical encoder	20
LTC 2622 Dual 12-Bit DACs	5
Total	60

It should be noted that the educational platform also provides a serial peripheral interface (SPI) to communicate

with other devices. For example, a LTC-2622 can be used to provide dual 12-bit D/A channels for signal monitoring on a digital oscilloscope. Figure 2 shows that the actual rotor position and the control command can be monitored at the same time on a digital storage oscilloscope.

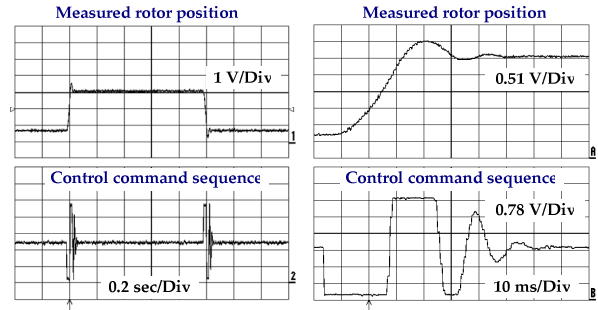


Fig.2 The experimental results of digital servo control of DC motors observed on a digital oscilloscope

2.2 Software

To help students quickly get familiar with the educational platform and know how automatic control algorithms are implemented using microcontrollers, sample software codes for the educational platform are given to the students in the hands-on laboratory. The sample software codes include the followings:

1. Interrupts implemented by using timers and their roles as clock sources in digital control algorithms;
2. USB communication and the protocol devised for the educational platform to send data to and receive commands from the PC;
3. A monitor and control software running on the PC side shown in figure 3 to collect experimental results from and send commands via universal serial bus (USB) to the educational platform;

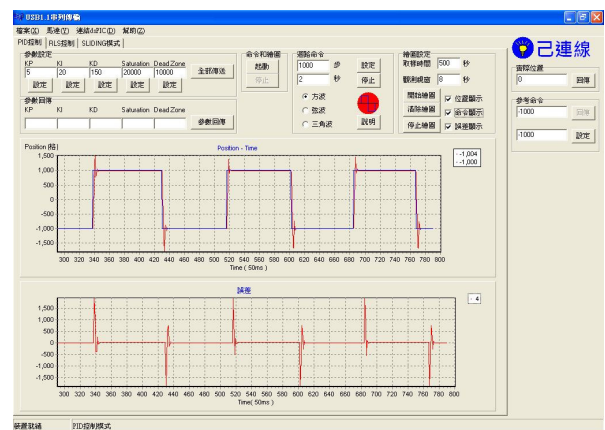


Fig. 3. A control and monitor software for the educational platform

4. Analog to digital conversions and how they are used to get the temperature data of the environment or the speed and position data of a dc motor;

5. Digital to analog conversions, serial peripheral interface and how they are applied to send out internal variables of the automatic control algorithms for monitoring;
6. Theory of pulse width modulation and its implementation;
7. Digital I/Os of the dsPIC microcontroller, and their application to use the HCTL-2021 (AVAGO technologies, 2006) to read the motor position;
8. The use of built-in encoder interface;

3. THE HANDS ON LABORATORY

The hands-on laboratory devised in this paper is used to lead students step by step not only how to use the educational platform and to implement their own control algorithms in microcontrollers, but also to know the basics of automatic control theory and practical implementation issues.

3.1 The digital signal microcontroller

The first part of the hands-on laboratory is used to remind students of those functions provided by most micro-processors, like interrupts, serial port communications, A/D conversions, pulse width modulation, and control of digital I/O ports. For example, the students should know in this part of the hands-on laboratory how to determine the period of the timer interrupt, and where to put the reaction codes once the timer interrupt is activated. The necessary theoretical backgrounds for A/D and D/A conversions, pulse-width modulations, and those programming skills for numerical differentiation and integration, etc. will also be reviewed for the students.

3.2 The modelling and simulation of dc motors

The students would be furthermore impressed about how mathematical models are used to predict the dynamical system behaviors at this part of the hands-on laboratory. The example used is the relationship between the input voltage and the output angular velocity of a brushed dc motor. The electrical characteristics in (1.a) of a brushed dc motor describe how the input voltage v_a is related to the armature current i_a and the output angular velocity ω . The mechanical dynamics in (1.b) shows how the armature current, the viscosity, and the load affect the angular velocity ω . The mathematical equations in (1.a-b) can then be combined in a block diagram shown in figure 4 (Franklin *et al*, 2002), and a SIMULINK behavior model in figure 5 with the parameters provided by the manufacturer can be constructed accordingly to predict the practical behaviors of a maxon s2332 dc motor. The dialog window shown in figure 6 is used to simplified the parameter setting process for the corresponding SIMULINK behavior model. The graphical user interface and the dialog window in SIMULINK environment are proved to be quite user friendly tools for the students.

$$L_a \frac{di_a(t)}{dt} + i_a(t)R_a + k_b \omega(t) = v_a(t), \quad (1.a)$$

$$J \frac{d\omega(t)}{dt} = k_e i_a(t) - b\omega(t) - T_L. \quad (1.b)$$

The experimental and simulation results of a brushed dc motor (maxon s2332) between the input voltage and the output angular velocity in figure 7 agree with each other very well. This also guarantees the students the validity of the mathematical model in (1.a-b) to predict the dynamical behaviors of a brushed dc motor.

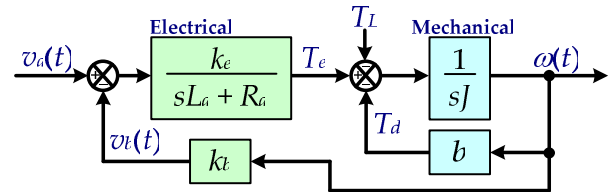


Fig.4 The block diagram for a brushed dc motor

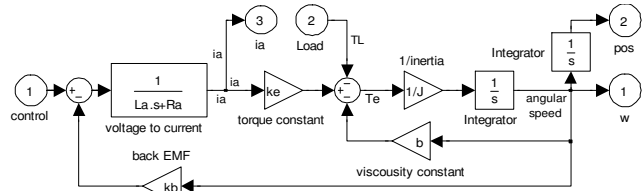


Fig.5 The SIMULINK behaviour model for a dc motor

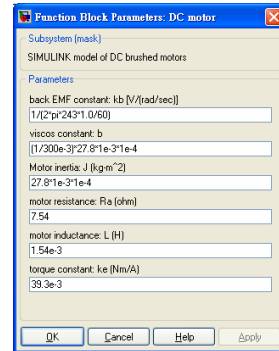


Fig.6 The dialog window used to set the brushed dc motor parameters in the SIMULINK environment

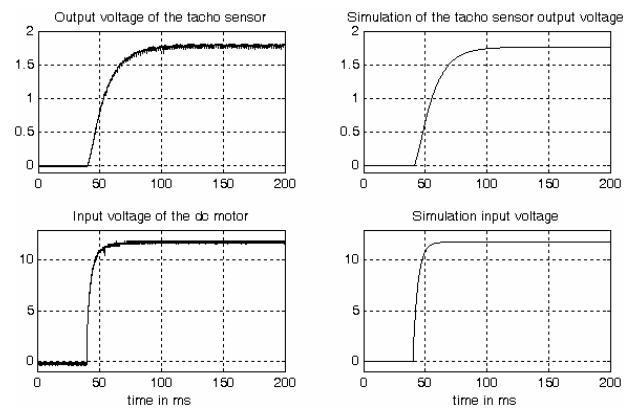


Fig. 7. The experimental and simulation results of the tachosensor output voltage of the maxon s2332 dc motor

3.3 The PID feedback controller design and simulation

The basic PID controllers with anti-windup functions and their implementation in digital form (Åström *et al* 1997, Bohn *et al* 1995) are introduced at this part of the hands-on laboratory. The students would first learn how to use encoders and tachometers to obtain motor positions and speeds signals. The structure of PID feedback controller with anti-windup function in figure 8 is then used as a starting point for the students to learn the basics of feedback controller theory and practical limitations in implementation.

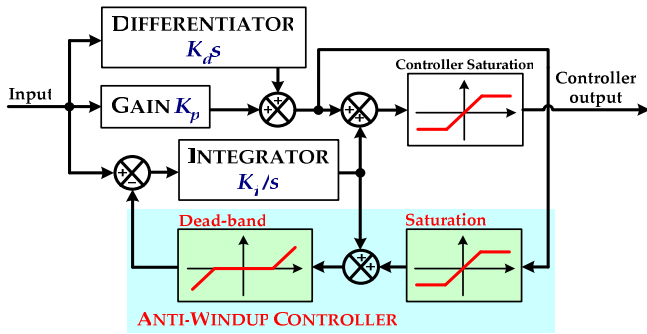


Fig. 8. A PID feedback controller with anti-windup function

The SIMULINK behavior model for a brushed dc motor obtained in the first part of the laboratory will also be used as a tool to help students learn quickly the effects of the proportional gain (P), the integral gain (I), the differential gain (D) and controller output saturation on the closed-loop system behaviors. For example, figure 9 illustrates the impacts of controller output saturation on the input command and output response. The results presented by Kelly *et al* (2001) can also be used to study different structure of PID controllers.

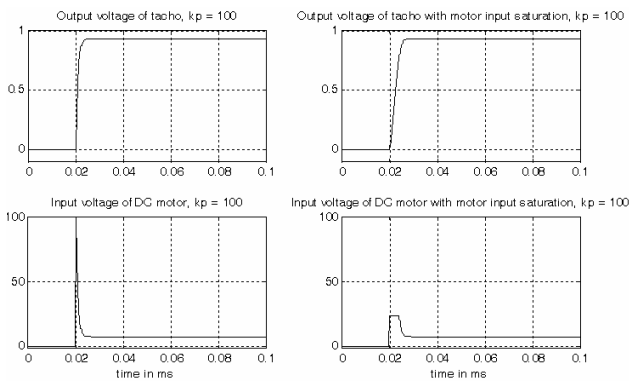


Fig. 9. The effect of controller output saturation on the dc motor speed control

The behavior model can also be used to make students acquainted with the digital form of the corresponding PID controller, and the effect of sampling frequency on the output responses. Figure 10 shows a SIMULINK behavior model used to illustrate the differences between continuous and discrete time PID controllers. The comparisons of output responses are also shown in figure 11 when the sampling frequency is set as 200Hz. The students will also learn how anti-windup function is used to eliminate the

influences of feedback controller output saturation on the closed-loop system performances.

After the students become familiar with operation of the PID controller in figure 8, they would be asked to implement the corresponding software codes into the educational platform in figure 1 at this part of the hands-on laboratory. The students should also check whether or not the experimental results are consistent with the theoretical predictions about steady state and transient responses. The sample codes listed in Table 2 are given to the students as references about the digital form of PID controller with anti-windup function which are written according to the signal flow in figure 8.

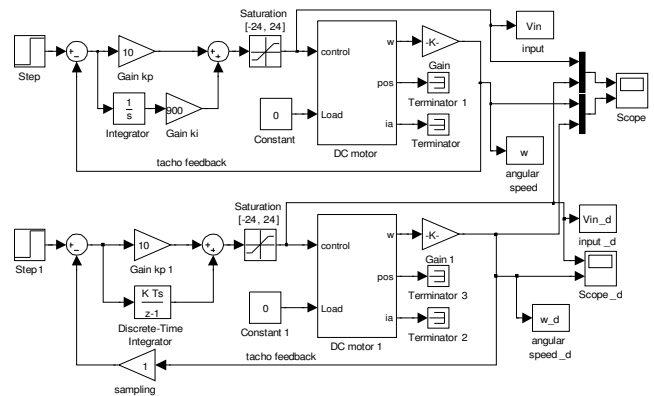


Fig. 10. A SIMULINK behavior model to show the differences between continuous and discrete time PID controllers

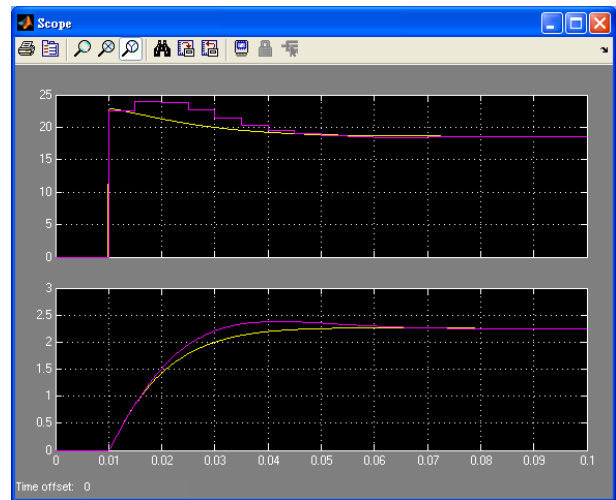


Fig. 11 The comparisons of continuous and discrete time PID controllers in the SIMULINK environment

Table 2. Sample C codes for a PID controller with the anti-windup function

```

/* sample C code for PID controllers with
   antiwindup function */
now_position = give_positon();
ek_1 = ek; //save previous error
//current error
    
```



```

ek = delta_command - now_position;
d_out = (ek-ek_1)*kd/100;
p_out = ek*kp)/100;
//integration of error with saturation
sk = sk_1 + uk - dead_zone((saturation(d_out +
p_out,saturation_parameter) + sk_1),
dead_zone_parameter); sk_1 = sk;
//PID controller with command saturation
yk = p_out + (sk*ki)/Fs + d_out;
yk = saturation(yk,117)*dir_flag;
//calculate and output PWM value
pwm_vol = 255-(yk+128); outp(pwm_vol,OCR0);
/** read current position ****
long give_positon(){
    long encoder_l_now, encoder_l;
    QEI_Position = ReadQEI();
    // take care of overflow
    encoder_l_now =0x0000ffff & QEI_Position;
    encoder_l += (encoder_l_now
-
encoder_l_old);
    // encoder_l_old is a global variable
    if((encoder_l_now-encoder_l_old) > 30000 )
        encoder_l -= 65536;
    else if((encoder_l_now-encoder_l_old)<-
30000)
        encoder_l += 65536;
    encoder_l_old=encoder_l_now;
    return (encoder_l);
}
/**saturation subroutine *****
long saturation(long in_value,long limit){
    if(in_value>=limit) {return limit;}
    else if(in_value<=(-limit))
        {return (-limit);}
    else {return in_value;} }
/** deadzone subroutine *****
long dead_zone(long in_value, long limit){
    if (in_value >= limit)
        {return (in_value - limit);}
    else if(in_value <= (-limit))
        {return in_value + limit;}
    else{return 0;} }
    
```

3.4 The fuzzy controller design of furuta pendulum

The abovementioned parts of the hands on laboratory introduced single input and single output automatic control systems to the students by using examples of brushed dc motor controls. The students would know in this part of hands-on laboratory that there are still multiple inputs and multiple outputs systems to be explored. The example used in this part is the feedback control of a furuta pendulum (Pan *et al*, 2000) shown in figure 12. It is a single input and double outputs system. The dc servo motor is used to control both the position and angle of the pendulum. Although the PID controller can still be used to control the furuta pendulum, the fuzzy control theory is used in this example such that the students are exposed to a different type of feedback control theory.

3.5 The swing up control of furuta pendulum

A simple control approach to swing the furuta pendulum from the downward position up to within a pre-specified region such that the positioning controller can then keep the pendulum up right at a certain rotational position is presented to the students at this part of the hands-on laboratory. This problem is interesting that the system is purposely controlled to be unstable until the furuta pendulum is swung up to make the angle stay in a pre-specified region. The experimental results in figure 13 show that the furuta pendulum was swung up at first and then kept at the up right position. In the meanwhile, the motor position was kept at 0.

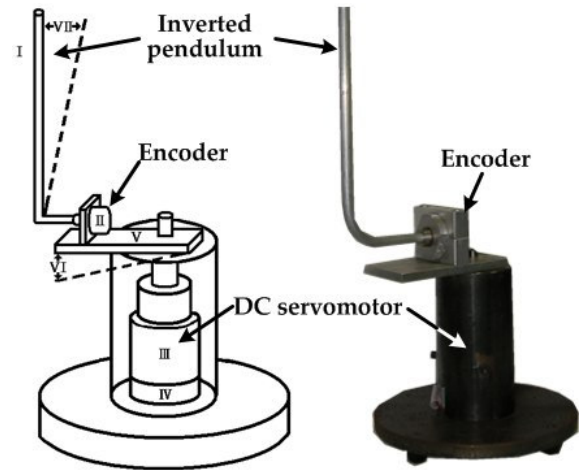


Fig. 12. The furuta pendulum system

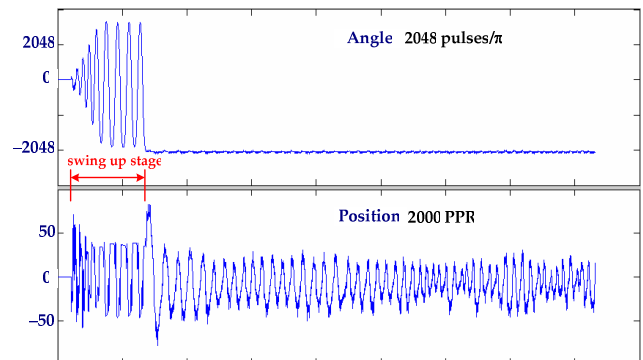


Fig. 13. The experimental results for a furuta pendulum system controlled by a fuzzy logic controller with swing up control

4. CONCLUSIONS

A hands-on laboratory for introductory automatic control courses is presented in this paper. The hands-on laboratory is devised based on a low cost and versatile educational platform. Therefore, the students can build their own platform, writing software control algorithms and check whether or not the system output responses coincide with those predicted by theory and behaviour model simulations. The students can learn better from doing practical experiments about automatic control theories and the related implementation issues. Although specific examples are used in the hands on laboratory, the automatic control theories

applied to achieve desired system performances is quite general.

REFERENCES

- Åström, K. J. and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd edition, Prentice-Hall, Inc., 1997.
- Avago Technologies, HCTL2021 Quadrature Decoder/Counter interface ICs, datasheet, Feb., 2006.
- Bernstein, D. S., "Enhancing undergraduate control education," *IEEE Control Systems Magazine*, pp. 40-43, October 1999.
- Bohn, C. and D. P. Atherton, "An analysis package comparing PID anti-windup strategies" *IEEE Control Systems Magazine*, pp. 34-40, April 1995.
- Chen, Y.-C. and J. M. Naughton, "An undergraduate laboratory platform for control system design, simulation, and implementation," *IEEE Control Systems Magazine*, pp. 12-20, June 2000.
- Franklin, G. F., J. D. Powell, and A. Mami-Naeini, *Feedback control of Dynamic systems*, 4th edition, Prentice-Hall, Inc., 2002.
- Kelly R. and J. Moreno, "Learning PID structures in an introductory course of automatic control," *IEEE Transactions on Education*, Vol. 44, No. 4, pp. 373-376, November 2001.
- Lee, C.-S., Juing-Huei Su, Y.-L. Wang, and J.-S. Fu, "An Open Courseware for Digital Control Systems," *the 2006 CACS Automatic Control Conference*, Taiwan, pp. 674-679, 2006.
- Linear Technology, LTC 2622: Dual 12-bit rail-to-rail DACs in 8-lead MSOP, www.linear.com.
- Microchip Technology, Inc., "Servo Control of a DC-Brush Motor," application note AN532, 1997.
- Pan, Yaodong, K. Furuta, S. Suzuki, and S. Hatakeyama, "Design of variable structure controller – from sliding mode to sliding sector," *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, pp. 1685-1690, Dec., 2000.
- Parallax, Inc., Basic Stamp microcontrollers, <http://www.parallax.com>.
- Parallax, Inc. Basic Stamp FAQ, <http://www.parallax.com/dl/docs/prod/stamps/BasicStampFAQ.pdf>, 2000.
- Tornambè, A., "Analysis and synthesis in the Simulink environment of control laws for dc motors: a real-time implementation with the microcontroller BASIC stamp II as a simple tool for an education control laboratory," *International Journal of Engineering Education*, vol. 21, no. 5, pp. 814-837, 2005.
- Urasaki, N., T. Senjyu, K. Uezato, and T. Funabashi, "Adaptive dead-time compensation strategy for permanent magnet synchronous motor drive," *IEEE Transactions on Energy Conversion*, Vol. 22, No. 2, pp. 271-280, June 2007.