

Velocity Feedback Experiments

<https://doi.org/10.3991/ijep.v7i1.6143>

Chiu H. Choi
University of North Florida, Florida, U.S.A.
cchoi@unf.edu

Abstract—Transient response such as ringing in a control system can be reduced or removed by velocity feedback. It is a useful control technique that should be covered in the relevant engineering laboratory courses. We developed velocity feedback experiments using two different low cost technologies, viz., operational amplifiers and microcontrollers. These experiments can be easily integrated into laboratory courses on feedback control systems or microcontroller applications. The intent of developing these experiments was to illustrate the ringing problem and to offer effective, low cost solutions for removing such problem. In this paper the pedagogical approach for these velocity feedback experiments was described. The advantages and disadvantages of the two different implementation of velocity feedback were discussed also.

Keywords—velocity feedback; microcontroller applications; tracking control; position control

1 Introduction

Position control problems are basic control problems in electrical, mechanical, and other engineering disciplines. The solutions to these problems should be covered in appropriate engineering curricula. Proportional control is one of the solutions to position control problems. The proportional gain determines whether the output response of a position control system to be fast or slow. A large proportional gain will produce a fast response with short rise time, which is generally desirable. But it may induce ringing, which is usually undesirable. Ringing is a transient response that the process variable overshoots and undershoots repeatedly its set point until the process variable settles at the steady state. In general, the larger the proportional gain, the higher the amplitude of the overshoot and undershoot in the ringing.

In this paper, we describe the experiments previously developed to observe ringing due to large proportional gain and compare two implementations of velocity feedback for suppressing ringing. The first implementation was operational amplifier based and the second was microcontroller based. Both of them are technologies at low cost. These experiments can be incorporated into microcontroller applications or control systems laboratory courses. The effort of developing control system experiments has been continual. Recent examples of such efforts include [1] through [5]. Few of these cover velocity feedback.

The rest of this paper is organized as follows. The pedagogical approach for the ringing problem and its solution by velocity feedback is described in Section 2. A comparison of these two implementations of velocity feedback is elaborated in Section 3. Concluding remarks are provided in Section 4.

2 Pedagogical Approach for the Ringing Problem and Solution

The ringing problem was illustrated in a plant, which is a Feedback Mechanical Unit Model 33-100 [6]. A simplified block diagram of this plant is shown in Fig. 1. This plant contains a position control system and a proportional controller to be designed by the user. The input shaft angle and the output shaft angle in Fig. 1 are illustrated in Fig. 2 (taken from [6]). Fig. 2 is a picture of the front panel of the Feedback Mechanical Unit Model 33-100.

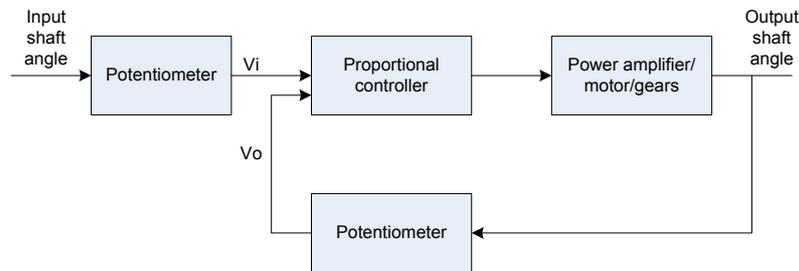


Fig. 1. Block diagram of the position control system

The two potentiometers shown in Fig. 1 are located behind the front panel in Fig. 2. These two potentiometers are identical. These potentiometers are used as transducers that convert the angular positions of the input and output shafts to electrical signals. The potentiometer outputs, V_i and V_o in Fig. 1 are in the range of approximately -10 V to +10 V. The Feedback Mechanical Unit Model 33-100 also contains a power amplifier that drives the output shaft through a DC motor and the associated belts and gears.

2.1 Operational amplifier implementation of velocity feedback

Proportional controllers can be built with various technologies, e.g., operational amplifiers or microcontrollers. A proportional controller was built with operational amplifiers for the experiments described in this paper. It was used to drive the output shaft of the Feedback Mechanical Unit to track the position of its input shaft. The proportional controller is shown in Fig. 3. There is no velocity feedback in the circuit. The output voltage of the proportional controller in Fig. 3 is given by

$$V_{out} = \frac{R_f}{R_1} (V_i - V_o)$$

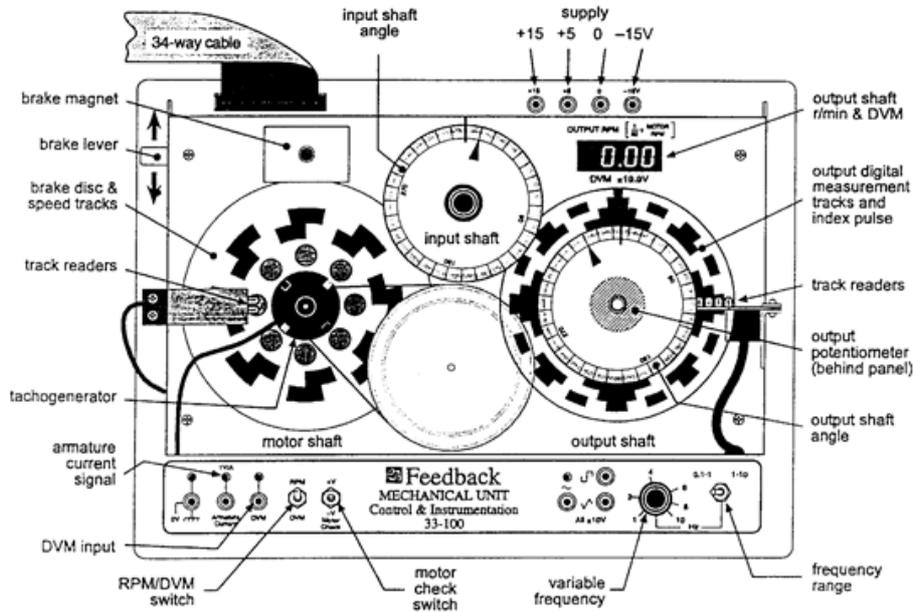


Fig. 2. Feedback Mechanical Unit Model 33-100

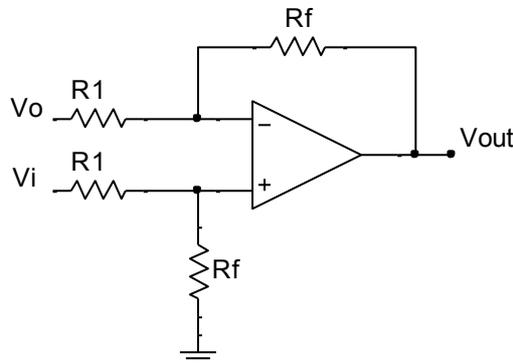


Fig. 3. Proportional controller

If the proportional gain (R_f/R_i) is large, ringing can occur. This is illustrated in Fig. 4 below. The square wave in green was the input shaft potentiometer signal V_i . The wave in yellow was the output shaft potentiometer signal V_o , which was eventually driven to be in alignment with the input shaft potentiometer signal after ringing was dissipated. In this experiment, the proportional gain R_f/R_i was set to 10 with $R_f=1\text{ M}\Omega$ and $R_i=100\text{ K}\Omega$. No velocity feedback was used.

Notice that in Fig. 4 the input shaft potentiometer signal (in green) was jagged during the period that the output shaft potentiometer signal (in yellow) was in the transi-

ent phase. That was the period that the output shaft was spinning to align itself to the position of the input shaft. Complete alignment occurred in the steady state is indicated in the red signal in Fig. 4. During that period, the output shaft was not spinning and the input shaft potentiometer signal was not jagged anymore.

The jagged portions of the green square wave signal were supposed to be smooth like the rest of the signal. Due to interference from the motor driving the output shaft, the smooth pulses became jagged during the transient phase. The interference was possible because the square wave generator and the motor shared the same power supply. The jagged portions of the green square wave signal could have been eliminated had a separate power supply been used solely by the square wave generator. But in this experiment, the power supply of the square wave generator was shared with the motor so as to educate the students about motor interference with function generator circuits.



Fig. 4. Output response with ringing (op amp-based controller)

A tachometer was connected to the output shaft for angular speed measurement. For clockwise rotation, the tachometer generated a positive voltage in the range of 0 to 5 V DC proportional to the angular speed of the output shaft. The higher the angular velocity, the higher the tachometer voltage. For counter-clockwise rotation, the tachometer worked the same way except that it generated a negative voltage in the range of 0 to -5 V DC. The tachometer factor was approximately 2.5 V DC per 1000 rpm.

The output shaft angular velocity was measured by the tachometer during the transient phase. The tachometer signal is shown in blue in Fig. 5. The interference observed in the green square wave (jagged green signal) coincided with the period that the tachometer signal was fluctuating. The interference disappeared when the tachometer voltage became zero steadily (motor not moving). This was another indication that the interference was caused by the motor.

In this experiment, the students could also observe that the blue tachometer signal in Fig. 5 was zero at either the local maximum or local minimum of the yellow output shaft potentiometer signal. This was anticipated because the blue tachometer signal should be proportional to the derivative of the yellow output shaft potentiometer signal. At these zero-crossing points of the tachometer signal, the output shaft angle increased (or decreased) no more and started reversing the direction of rotation.

A method to remove ringing is by using the tachometer signal in negative feedback. The block diagram for this velocity feedback method is shown in Fig. 6. In that block diagram the block for the proportional controller with velocity feedback was realized as a single operational amplifier circuit as shown in Fig. 7. The tachometer voltage V_{tach} in Fig. 6 and 7 was a signal between -5 V to +5 V in this experiment. The output voltage V_{out} in Fig. 7 is given by

$$V_{out} = \frac{R_f}{R_1}(V_i - V_o) - \frac{R_f}{R_2}V_{tach}$$

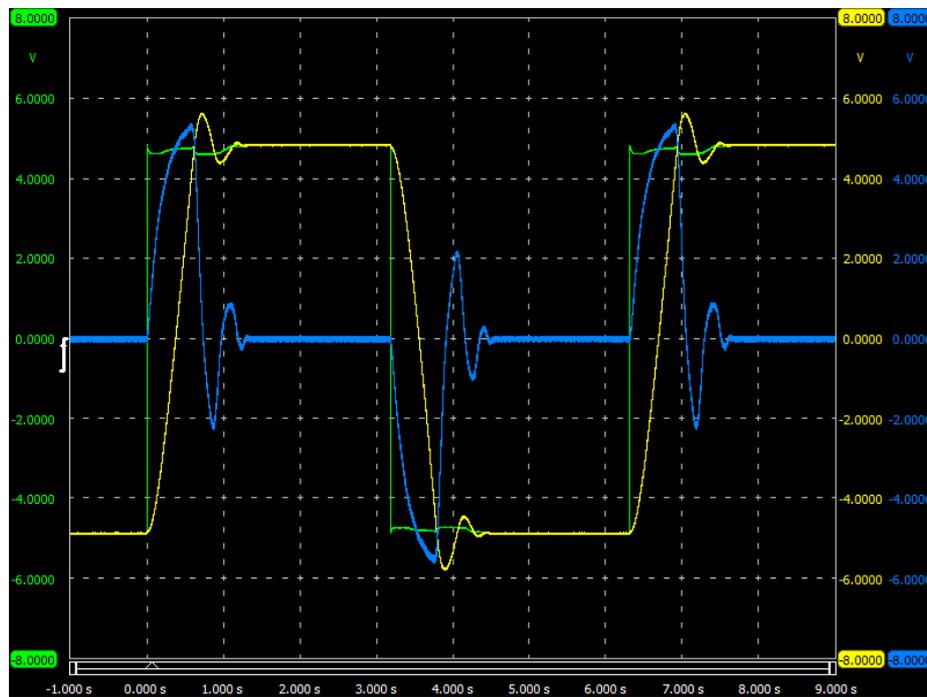


Fig. 5. Tachometer signal (in blue)

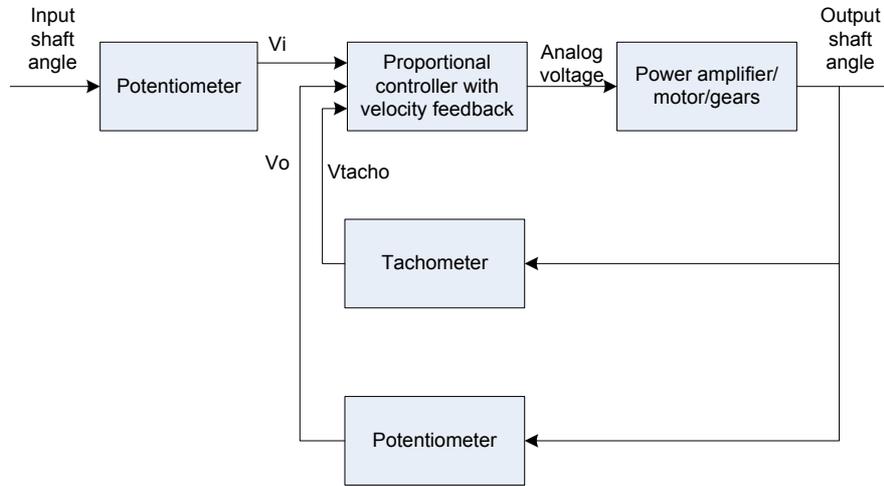


Fig. 6. Position control system with operational amplifier-based velocity feedback controller

This circuit in Fig. 7 was used to control the output shaft with the proportional gain $R_f/R_I=10$ again. The velocity gain was chosen to be $R_f/R_2=5$. The values of the resistors were $R_f=1\text{ M}\Omega$, $R_I=100\text{ K}\Omega$, and $R_2=200\text{ K}\Omega$. The input and output shaft potentiometer signals were recorded in Fig. 8. Notice that the output shaft potentiometer signal eventually aligns with the input shaft potentiometer signal but without ringing. The velocity feedback removed the ringing. Various methods in the literature, e.g., root locus method, can be used to find the proportional and velocity gains so that the ringing will not occur in the output signal. These methods are readily covered in the literature, e.g., [7] and [8] and are not discussed in this paper.

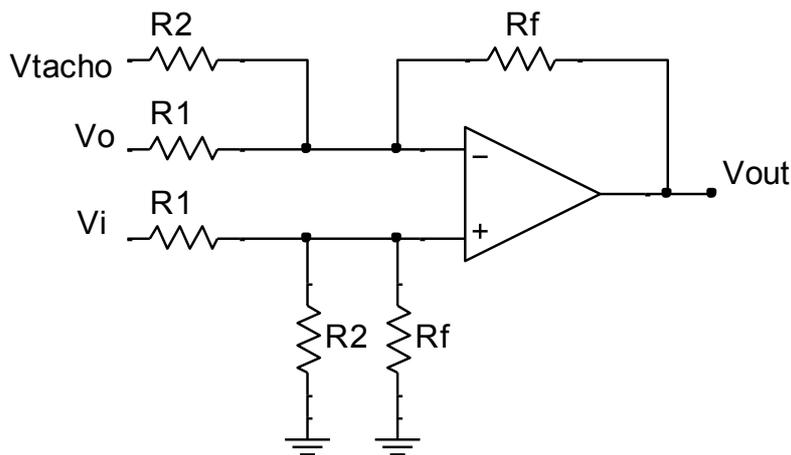


Fig. 7. Proportional controller with velocity feedback



Fig. 8. Output response with ringing removed by velocity feedback (op amp-based controller)

2.2 Microcontroller implementation of velocity feedback

In this second experiment the proportional controller with velocity feedback was implemented as a C program that ran on a Freescale (now NXP) 16-bit microcontroller, model MC9S12C32 in the S12 family. A block diagram for this microcontroller approach is shown in Fig. 9. The software development tool used was CodeWarrior Development Studio for HCS12(X) Special Edition 5.1. Users can use it to develop firmware in C, assembly, and some other languages. The microcontroller and the software development tool were discussed in [3].

In the Feedback Mechanical Unit Model 33-100, the output shaft is driven by a motor through a gear train. The motor can also be controlled by a locked-antiphase pulse-width modulated (PWM) signal through an H-bridge. The working of it is explained as follows: if the duty cycle of the PWM signal is less than 50%, the output shaft will move in the clockwise direction. The smaller the duty cycle, the more the power is delivered to the motor to move the output shaft. If the duty cycle is greater than 50%, the output shaft will move in the counter-clockwise direction and the higher the duty cycle, the more the power is delivered to the motor. The neutral point is at 50% duty-cycle, which turns off the motor. This locked-antiphase approach was used in [3] before.

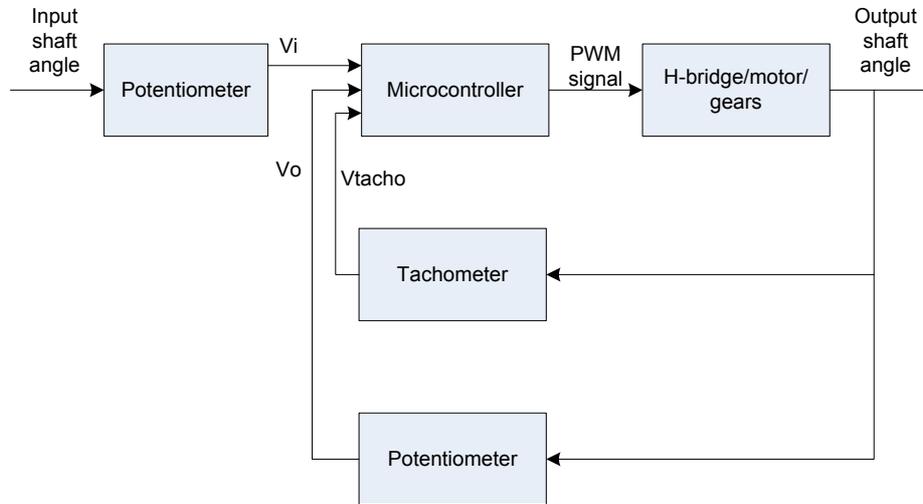


Fig. 9. Position control system with microcontroller-based velocity feedback controller

The pseudo code for implementing the proportional controller with velocity feedback for this experiment is described as follows:

1. Use the microcontroller's on-chip analog-to-digital converter to read the voltages of the input and output shaft potentiometers, and the tachometer.
2. Calculate the new PWM duty cycle for controlling the output shaft as a function of position error and the angular velocity of the output shaft. Velocity feedback is incorporated into the PWM signal.
3. Scale the duty cycle and limit it to be within 0 to 100%.
4. Use the microcontroller's on-chip pulse-width modulator to generate the PWM signal at this new duty cycle. Operate the motor for 50 ms before starting over the same process again.
5. Loop back to step 1.

The code snippet of the C program that implemented the proportional controller with velocity feedback is shown below (next page).

```

/*****
Proportional controller with velocity feedback
*****/
while(1) {
// ADC conversion sequence
ATDCTL5=0x10; // start conversion
Vi=ATDDR0H; //Input shaft position
Vo=ATDDR1H; //Output shaft position
Vtacho=ATDDR2H; //speed of motor
// calculate the new PWM duty cycle
n=-(vin-vout)/255/2*Kp+(vtacho-128)/ 128/2*Kv +
pwdty;
// scaling and limiting the duty cycle to within 0 to
100%
n=n*250; //scaling n
if (n>250) n=250;
else if(n<0) n=0;
else if ((n>110)&(n<=120)) n=110;
else if ((n>=130)&(n<140)) n=140;
//update duty cycle and operate for 50 ms
PWMDTY0= (unsigned int) n;
delayms(50);
} //end of while loop

```

The on-chip pulse-width modulator was initialized as shown in the code snippet below. The PWM period counter, `PWMPER0`, was set to 250. The PWM duty cycle counter, `PWMDTY0`, determines the duty cycle. The formula for the duty cycle of the PWM signal is equal to the ratio of the value stored in `PWMDTY0` to that stored in `PWMPER0`.

```

/*****
INITIALIZATION OF PWM CHANNEL 0
*****/
void init_PWM(void) {
PWMPRCLK=0x06; //Set Clock A to 125 KHz
PWMCLK=0x00; // PWM clock at 125 KHz
PWMCTL=0x00; //8 bit resolution
PWMPOL=0x01; //Channel 0 is active high
PWMCAE=0x00; // Channel 0 left aligned
PWMPER0=250; // PWM period=2ms (500 Hz)
PWMDTY0=125; //Duty cycle 50% initially
PWME=0x01; // Enable channel 0
}

```

The C program that implemented the proportional controller with velocity feedback was tested on the Feedback Mechanical Unit Model 33-100. In the first test, only

proportional gain ($K_p=25$) was used and no velocity feedback ($K_v=0$) in the first code snippet above. The output shaft potentiometer signal is shown in yellow in Fig. 10. Notice that there was ringing



Fig. 10. Output response with ringing (microcontroller-based controller, $K_p=25$ and $K_v=0$)

Next, the velocity feedback gain was changed from 0 to 2 ($K_v=2$) and the proportional gain remained unchanged ($K_p=25$). The output shaft signal is shown in yellow in Fig. 11. Notice that the ringing was greatly reduced. This indicated that the microcontroller-based implementation of velocity feedback was effective also in suppressing ringing.

In this microcontroller approach, signal conditioning circuits for interfacing the three analog-to-digital converter channels to the input potentiometer, output potentiometer, and the tachometer were required. The signal conditioning circuits for the input and output potentiometers were identical. They converted the potentiometer voltage in the range of -10 V to +10 V to the range of 0 to 5 V in order to meet the electrical characteristics of the microcontroller. A solution for such signal conditioning circuit is shown in Fig. 12. Notice that only one operational amplifier is used. The equation for the output voltage is given by

$$V_{out} = 0.25V_i + 2.5,$$

where V_i and V_{out} are the input and output of the signal conditioning circuit, respectively.

Another signal conditioning circuit is required to convert the tachometer voltage from -5 V to +5 V to the range of 0 to 5 V. The circuit diagram is shown in Fig. 13. The equation for the output voltage is given by

$$V_{out} = 0.5V_{tachometer} + 2.5.$$

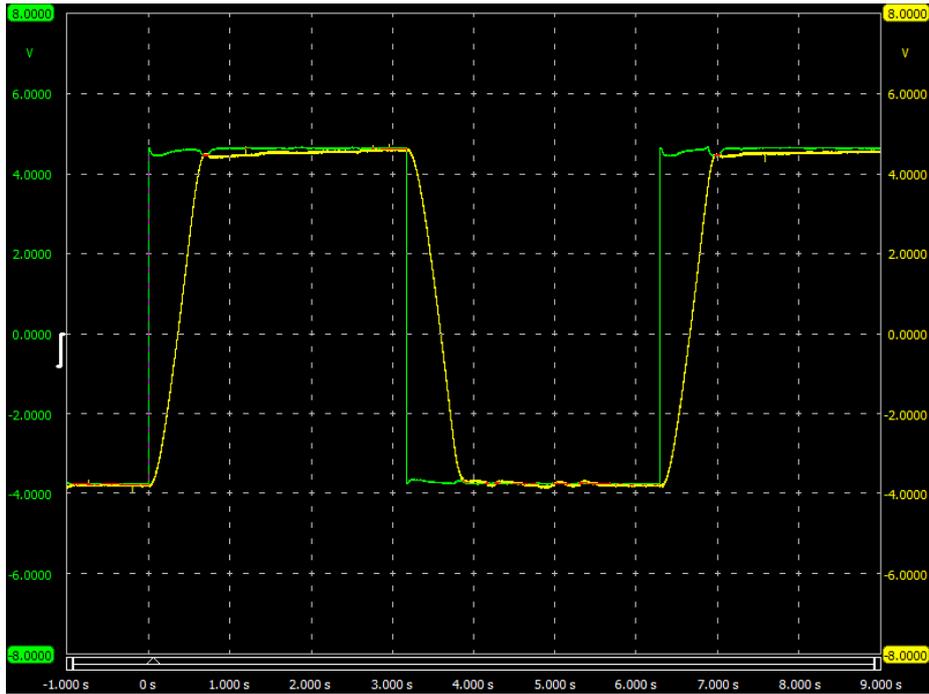


Fig. 11. Output response without ringing by using velocity feedback (microcontroller-based controller, $K_p=25$ and $K_v=2$)

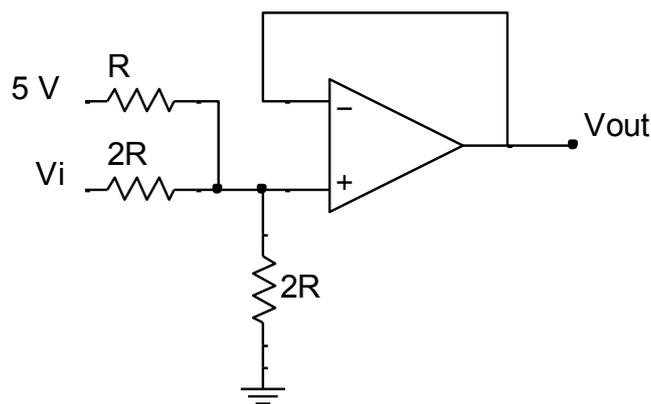


Fig. 12. Signal conditioning circuit for input and output shaft potentiometers

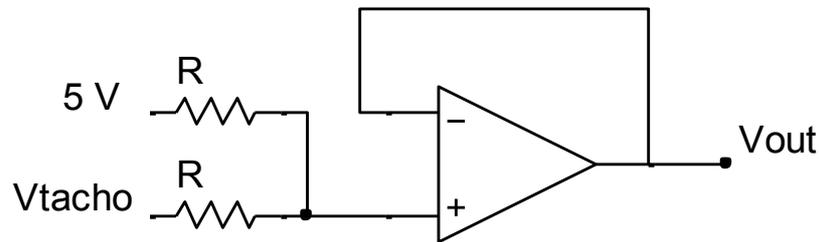


Fig. 13. Signal conditioning circuit for the tachometer

3 Comparison of the Two Implementations

The two implementations described in the last section were compared against each other. The following aspects were compared: cost, time-to-finish, design tools, and re-configurability. These aspects are those that the students should understand in their future work on the same topic. On the comparison of the cost, the operational amplifier approach was less expensive because it required only one operational amplifier and a few resistors as indicated in Fig. 7. Whereas the microcontroller approach required signal conditioning circuits (two copies of Fig. 12 and one copy of Fig. 13) and a microcontroller module and the associated software development tools. The cost of the microcontroller implementation was higher than the operational amplifier implementation.

On the comparison of the time-to-finish, the operational amplifier implementation was much simpler to build and took much less time because the circuit was simple. The microcontroller implementation took more time because there were three signal conditioning circuits to build and there was a C program to write and debug, which could be time consuming depending on the programming skills of the students.

On the comparison of the design tools, the operational amplifier implementation required few tools. A datasheet and an operational amplifier textbook for reference purposes were sufficient. The microcontroller implementation required more design tools for developing the C program. While most of these tools are efficiently bundled together in the integrated development environment, usually there is still a significant learning curve before becoming proficient in using these tools.

On the comparison of the re-configurability, for example, changing the proportional and velocity gains, the operational amplifier implementation required the replacement of resistors in the circuit. This is usually tedious in those cases that the resistors were soldered on to the printed circuit board. For the microcontroller implementation, that can be accomplished conveniently in software. Changing the gain requires changing the numeric values of the variable in the C program and re-compiling the program. The integrated development environment has made such process very simple. Further, additional features can be programmed into the controller C program easily.

4 Concluding Remarks

The operational amplifier-based and the microcontroller-based velocity feedback controllers were compared. Both were effective in suppressing ringing. The operational amplifier implementation was simpler to prototype than the microcontroller implementation but harder to make changes to the proportional and velocity gains. The operational amplifier implementation was intuitively simpler to understand than the microcontroller implementation. Such approach is effective for delivering the concepts and workings of velocity feedback to the students at minimal cost and time. On the other hand, it is much easier to improve the performance of the microcontroller-based controller because it is done in the software environment without altering any hardware. However, it required more hardware, e.g., signal conditioning circuits than the operational amplifier approach. Another advantage of the microcontroller-based controller is that the code developed can be re-used in future projects. This can reduce the software development time. Knowing these advantages and disadvantages through these lab experiments, one can select the appropriate approach for their own applications.

5 References

- [1] R.M. Reck and R.S. Sreenivas, "Developing a new affordable DC motor laboratory kit for an existing undergraduate controls course," American Control Conference, Chicago, IL, 2015, pp. 2801-2806. <https://doi.org/10.1109/acc.2015.7171159>
- [2] C. Choi, "Comparing microcontroller and analog methods for tracking control experiments," Frontiers in Education Conference, El Paso, TX, 2015, pp. 1-4. DOI: 10.1109/FIE.2015.7344418. <https://doi.org/10.1109/FIE.2015.7344418>
- [3] C.H. Choi, "Microcontroller-based feedback control laboratory experiments," *International Journal of Engineering Pedagogy*, vol. 4(3), pp. 60-66, 2014. <https://doi.org/10.3991/ijep.v4i3.3529>
- [4] C. M. Ionescu et al., "A remote laboratory as an innovative educational tool for practicing control engineering concepts," *IEEE Trans. Edu.*, vol. 56, no. 4, pp. 436-442, 2013. <https://doi.org/10.1109/TE.2013.2249516>
- [5] M. Gunasekaran and R. Potluri, "Low-cost undergraduate control systems experiments using microcontroller-based control of a DC motor," *IEEE Trans. Edu.*, vol. 55, no. 4, pp. 508-516, 2012. <https://doi.org/10.1109/TE.2012.2192441>
- [6] 33-002 *Servo Fundamentals Trainer Manual*, Feedback Instruments Ltd..
- [7] N. S. Nise, *Control Systems Engineering*, 7th ed. John Wiley, 2015.
- [8] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 12th ed. Prentice Hall, 2010.

6 Author

Chiu H. Choi is a professor in the Department of Electrical Engineering at the University of North Florida, Florida, U.S.A. He earned his Master's and Ph.D. degrees in electrical and computer engineering at the University of California, Santa Barbara.

Article submitted 12 August 2016. Published as resubmitted by the authors 15 February 2017.