International Journal of Automotive Technology, Vol. 7, No. 3, pp. 369–375 (2006)

1229–9138/2006/027–19

DEVELOPMENT OF A SIMPLE CONTROL ALGORITHM FOR SWIRL MOTOR CONTROLLER

W. T. LEE[1)*] and J. J. KANG[2)]

[1)]Department of Control and Instrumentation Engineering, Changwon National University,
Gyeongnam 641-773, Korea
[2)]Department of Control and Instrumentation Engineering, Graduate School, Changwon National University,
Gyeongnam 641-773, Korea

(Received 25 October 2005; Revised 16 March 2006)

ABSTRACT–This paper describes a simple proportional and integral control algorithm for a swirl motor controller and its application. The control algorithm may be complicated in order to have desired performance, such as low steady state errors, fast response time, and relatively low overshoot. At the same time, it should be compact so that it can be easily implemented on a low cost microcontroller, which has no floating-point calculation capability and low computing speed. These conflicting requirements are fulfilled by the proposed control algorithm which consists of a gain scheduling proportional controller and an anti-windup integral controller. The mechanical friction, which is caused by gears and a return spring, varies very nonlinearly according to the angular position of the system. This nonlinear static friction is overcome by the proportional controller, which has a two-dimensional look up gain table. It has error axis and angular position axis. The integral controller is designed not only to minimize the steady state error but also to avoid the windup effect, which may be caused by the saturation of a motor driver. The proposed control algorithm is verified by use of a commercial product to prove the feasibility of the algorithm.

KEY WORDS : SMC (Swirl motor controller), Nonlinear friction, Gain scheduling proportional control, Anti windup integral control

1. INTRODUCTION

Swirl control system is a device to control the air mass flow by promoting mixing the fuel with air. High speed of engine allows the fuel to flow rapidly. Hence, the fuel can be mixed properly and the firing speed can be high. Contrast to this, under low engine speed the speed of mixed fuel flow to decrease. It is the swirl system that can make the fuel into cylinder mixed properly under low speed of engine.

In swirl control system, the intake port of engine, which lets an air into cylinder from air-absorbing port, is separated to two parts. And one of these parts is closed to product swirl easily by increasing the air-absorbing speed under low speed of the engine.

Swirl motor controller (SMC) is DC motor driver that provide the precious position of the swirl plate according to input of the engine control unit (ECU). The performance of the SMC determines the efficiency of the engine. So, the controller must suitably perform with low steady state error, fast response time, and relatively low overshoot.

Swirl control system, which this paper handles, has nonlinear mechanical friction due to the gears and a return spring. Therefore, the control algorithm of SMC should be developed focusing on the compensation of nonlinear friction. SMC consists of two kinds of controller. One is a gain scheduling proportional controller which overcomes nonlinear friction. And the other is an anti-windup integral controller which reduces steady state errors. Besides it is simple for operating low cost microcontroller that has no-floating calibration and low computing speed.

The control algorithm is developed by simulation using SIMULINK® and FIXED_POINT® (Fix-Pt®) toolbox of MATLAB®. SIMULINK® helps to represent control algorithm and to design control algorithm easily. Fix-Pt toolbox helps to convert floating-point into fixed-point easily and to detect operating error, overflow and saturation. The developed algorithm it is proved by experiments.

2. OPEN-LOOP CONTROL EXPERIMENT

The best way to know a system is making open-loop

**Corresponding author.* e-mail: wootaik@sarim.changwon.ac.kr

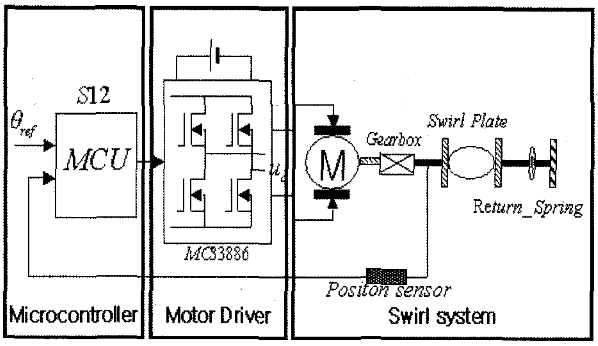


Figure 1. Block diagram of swirl control system.

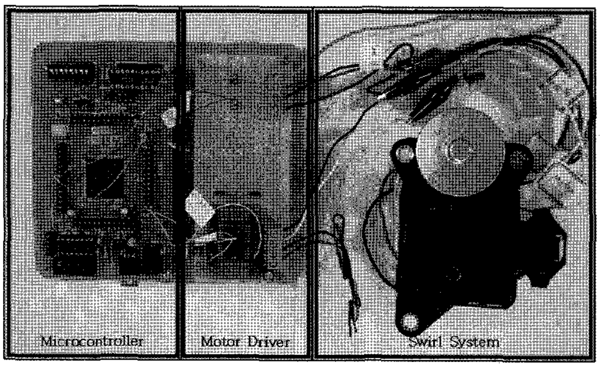


Figure 2. Photograph of the Swirl control system.

control experiments. Through this experiment, characteristics of the swirl system such as the position of saturation and moving range can be noticed and existence of the non-linear friction can be proved.

2.1. Architecture of Swirl Control System

Figure 1 shows the functional schematic of swirl control system and Figure 2 shows the real appearance of swirl control system. It can be separated to three parts, swirl motor controller, motor driver and swirl system.

The microcontoller of SMC is HCS12. It produces PWM signals and measures the position of swirl plate to be calculated by the control algorithm. Motor driver is a drive IC (MC33886), which supplies PWM power signals to control DC motor. The swirl system has a DC motor, potentiometer, gears and return spring

2.2. Open-loop Control Experiment

Figure 3 shows the result of open-loop control experiment. The horizon of graph means the variation of the input voltage and the verticality of graph indicates the variation of position voltage.

With the view of the result graph, the saturation point of output is the min 0° and the max 98°. Though the same input was given for the swirl system, different outputs were gotten on different position of swirl plate. This result is caused by non-linear friction (Deur *et al.*, 2004; Pechlaner and Björn, 2001).

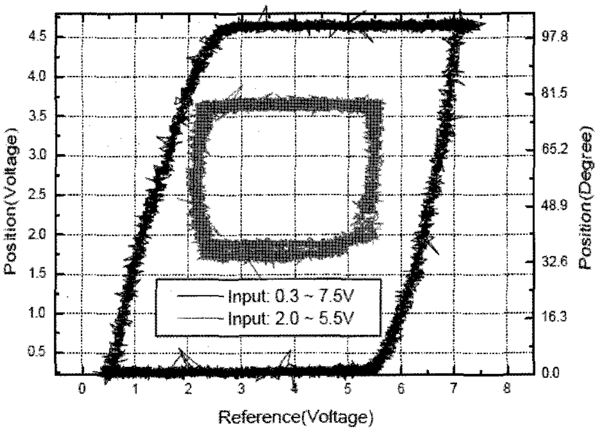


Figure 3. Input: 0.3–7.5 V, 2.0–5.5 V.

The friction of swirl system is divided into two elements, which are static friction and coulomb friction. When the swirl plate comes to an end, static friction is the force disturbing the beginning of movement. When the swirl plate is moving, coulomb friction is the force disturbing the movement. The total friction means addition of static friction and coulomb friction (William and Leroy, 1996). The output of system is different according to the position of system and change of input. It is proved by existence of non-linear friction.

3. DEVELOPMENT OF CONTROL ALGORITHM

The control algorithm is developed by simulation. MATLAB® supports Fix-Pt® toolbox and SIMULINK® that are very useful to develop the control algorithm. The Fix-Pt® toolbox has the advantage that it can monitor each SIMULINK® block data such as the number of overflow and the magnitude of min, max when the fixed-point arithmetic is operated (Yeo *et al.*, 2004).

3.1. Controller using Floating-point Calibration

Generally, the control algorithm is developed using floating-point arithmetic because it is easy to calculate and operate.

Through the simulation that would implement control algorithm has 2 controllers, a proportional and integral controller (Charles and Troy, 1995). Each controller output is calculated by gain (K_P, K_I) and error that subtracts position ($Y(k)$) from reference ($R(k)$) and used for different purpose. The proportional controller output ($V_P(k)$) added the integral controller output ($V_I(k)$) to make the controller output ($V_C(k)$) just as equation (1).

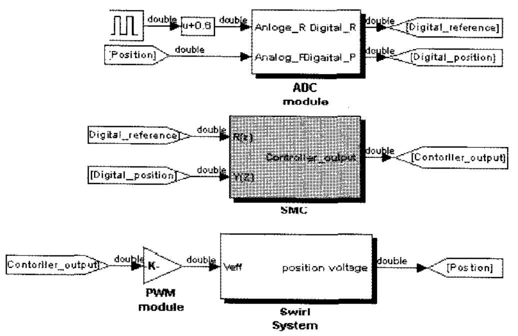
$$V_C(k)=V_P(k)+V_I(k) \tag{1}$$

The role of proportional controller is to compensate for non-linear friction (Kang *et al.*, 2005). As already explained, the friction of swirl system is different according to the position of system and change of input. Therefore the proportional gain is selected with gain-scheduling method using 2 dimensional look up table which has an error axis and an angular position axis. The proportional controller output is gotten by equation (2).

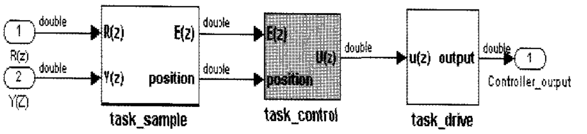
$$V_P(k)=K_P \times (R(k)-Y(k)) \tag{2}$$

The role of integral controller is to reduce steady state errors and guarantee low steady state errors when there are external disturbances. But there are some possibilities of being unstable system with oscillation called wind-up effect. For the reason, the system with saturation of actuator needs anti wind-up integral controller. The integral controller output is gotten by equation (3) with control period (T_S) and previous output ($V_I(k-1)$).

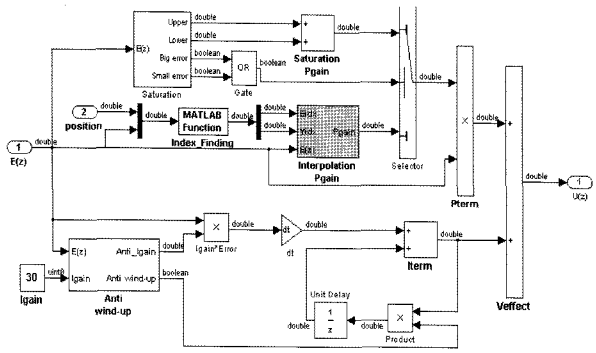
$$\begin{gathered} V_I(k)=((K_I \times (R(k)-Y(k)) \times T_S)+V_I(k-1) \\ -0.66^\circ \le (E(k)=(R(k)-Y(k)) \le 0.66^\circ \end{gathered} \tag{3}$$



(a) Whole system model with floating-point



(b) SMC model with floating-point



(c) Task control

Figure 4. Simulation SMC using floating-point arithmetic.

Figure 4(a) shows swirl control system with floating-point arithmetic. All data types of signal flow are the double which means floating-point arithmetic applying to all calculations. It is composed of 3 parts. They are the ADC module, SMC and swirl system block. First, the ADC module block that has 2^{10} resolutions is AD converter of HCS12. Second, swirl system block is the model of real swirl system and the motor driver gain block converts duty ratio of −100~100% range which is the output of SMC into −12V~12V. At last the SMC block executes the control algorithm.

Figure 4(b) shows the SMC block that consists of 3 parts which are task-sample, task-control and task-drive. First, task-sample executes to exchange ADC value with calibration value and calculating error. Moreover it transfers the calculated error and position value to task-control. Second, task-control implements the proportional controller using the gains scheduling method. Lastly, task-drive executes to calculate duty ratio using output of task-control.

Figure 4(c) shows the task-control block that includes the proportional and integral controller. In the first place, the proportional gain is selected by two ways. In the one way the saturation gain is chosen in case that the error is bigger than 2 or lower than −2. If a small gain is given to the system when a big error occurs, the system output has the overshoot and undershoot because the output of proportional controller is error multiplied with the gain. So this gain is the smallest value in gain table and constant. The other way is using linear interpolated gain of gain table due to the error and angular position through equation (4).

$$K_P=(((K_{Y\_E}-K_{Y\_E-1}) \div (E(k)-E_I))\times(E_I-E_{I-1}))+K_{Y\_E-1} \tag{4}$$

In equation (4), K_{Y_E} is the selected gain according to Yidx and Eidx, and K_{Y_E-1} is the gain according to Yidx and Eidx-1. E_I is the error of E-axis according to Eidx and E_I is the error of E-axis according to Eidx-1.

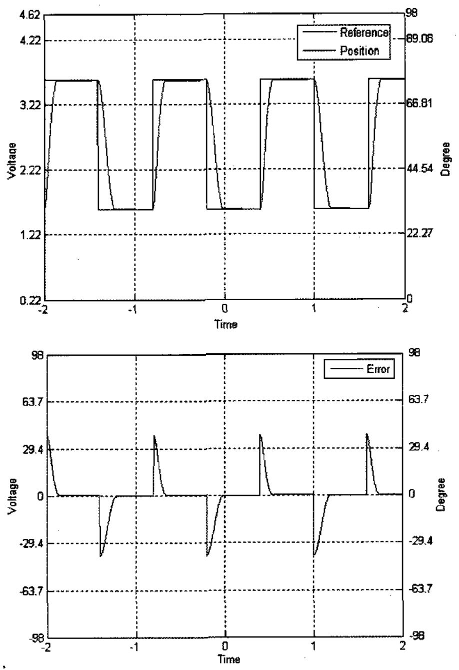
Next, the integral gain is selected by simulation, and the result is 30. This integral controller prevents wind-up effects from conducting the integral controller when the error is very small just like $-0.66^\circ \le E(k) \le 0.66^\circ$ in equation (3).

Table 1 is the result of gain look-up table owing to a repeat simulation. With the view of result, as the error gets smaller, steady state error is made to decline with bigger gain. In opposed cases decreased the gain prevents overshoots and undershoots. And the bigger friction of swirl system is created by the bigger angular position owing to return spring. If there is the bigger position angle, bigger gain is needed. Addition to this one more consideration was fast response time. The values of Y-axis and E-axis are properly chosen through repeated simulations.

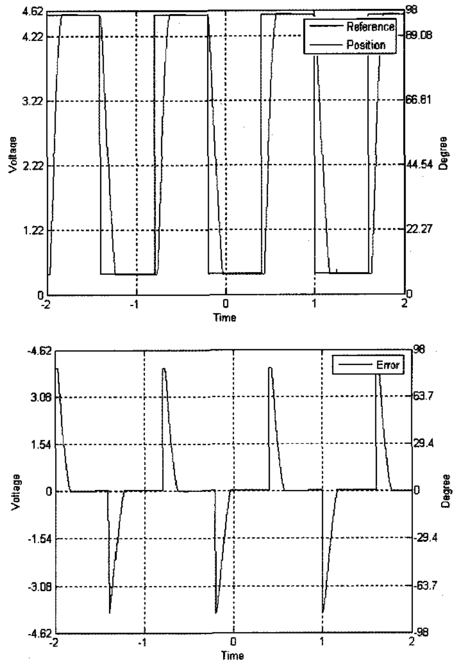
Table 1. Gain look up table with floating-point.

	E-axis	-44°	-6.6°	0°	6.6°	44°
Y-axis	Eidx Yidx	1	2	3	4	5
0°	1	1.8	7	28	9.5	2.5
22°	2	1.8	7	29	9.5	2.5
44°	3	1.8	7	30	9.5	2.5
66°	4	1.8	7	31	9.5	2.5
88°	5	1.8	7	33	9.5	2.5

Figure 5 describes the graph of simulation that makes use of table 1 verifying the performance of control algorithm. The control algorithm satisfies low overshoot, low steady state errors and fast response time according to the change of error and angular position. The control algorithm that utilizes floating-point arithmetic has good performance.



(a) Reference: 30 ~ 73°



(b) Reference: 10 ~ 97°

Figure 5. Controller performance using floating-point.

3.2. Controller using Fixed-point Calibration

Fixed-point is an alternative form for expressing numerical values. Fixed-point arithmetic is the integer arithmetic but it allows the fractions, and usage of the general low cost microcontroller that has no floating-point calibration and operates the arithmetic with integer number.

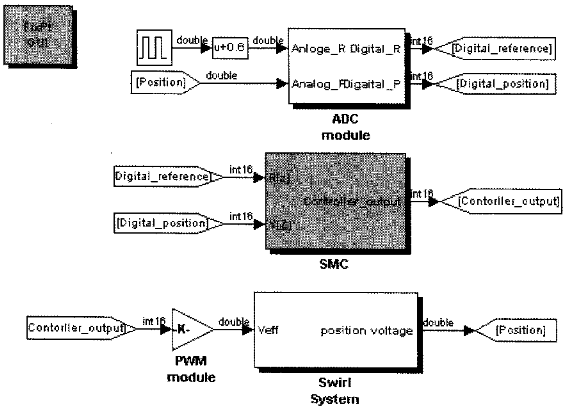
The integer doesn't express the fraction number. Hence, the developer should consider the decimal point (called radix point). The right side bits of decimal point are integer bits and the left side bits are the fraction bits. Therefore the one integer number has different meaning in fixed-point number. The decimal point does not exist exactly, but it exists only in idea of developer (Labrosse, 2000).

The developed control algorithm using floating-point arithmetic is converted into the algorithm of fixed-point arithmetic by FixPtR toolbox. The decimal point and bit size are important when floating-point is converted into fixed-point using FixPtR toolbox because it selects the range of number expressing the integer and fraction number. Due to the simulation result, every simulation block has proper size and decimal point using fixed-point toolbox. And every fixed-point number and operating result is verified by FixPt-GUI block.

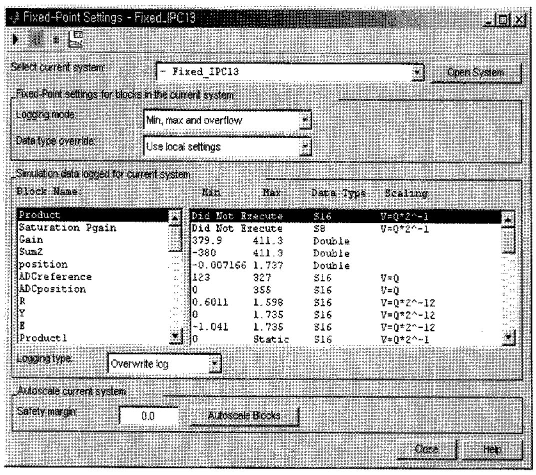
Figures 6(a), (c), (d) are the same with Figures 4(a), (b), (c) and play same roles. But they take advantage of fixed-point arithmetic. The data type of signal flow shows the decimal point and number size of each simulation block.

Figure 6(b) is FixPt GUI block that represents the overflow times, min, max values and scaling factors of each simulink block. It is very useful to know whether the decimal point is in proper location or not. Figure 6(e) is Interpolation P gain block that interpolates proportional gain toward E-axis.

When SMC model is simulated using Table 1, the control algorithm performance that makes use of fixed-point arithmetic is worse than the performance using



(a) Whole system model with fixed-point



Select current system: - Fixed_IPC13

Fixed-Point settings for blocks in the current system

Logging mode: Min, max and overflow

Data type override: Use local settings

Simulation data logged for current system

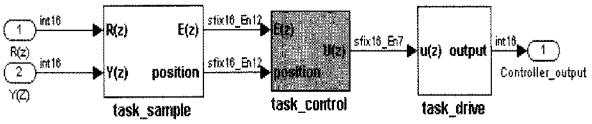
```
Block Name:             Min        Max     Data Type  Scaling
Product                 Did Not Execute    S16        V=Q*2^-1
Saturation Pgain        Did Not Execute    S8         V=Q*2^-1
Gain                    379.9      411.3   Double
Sum2                    -380       411.3   Double
position                -0.007166  1.737   Double
ADCreference            123        327     S16        V=Q
ADCposition             0          355     S16        V=Q
R                       0.6011     1.598   S16        V=Q*2^-12
Y                       0          1.735   S16        V=Q*2^-12
E                       -1.041     1.735   S16        V=Q*2^-12
Product1                0          Static  S16        V=Q*2^-1
```

Logging type: Overwrite log

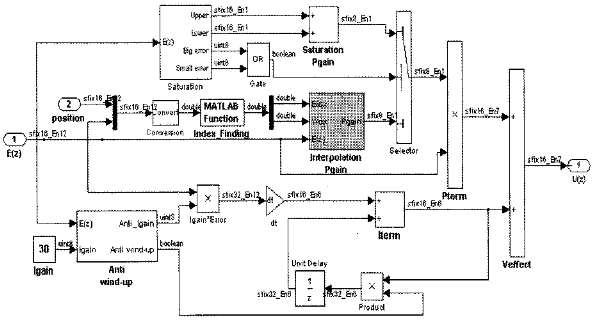
Autoscale current system

Safety margin: 0.0

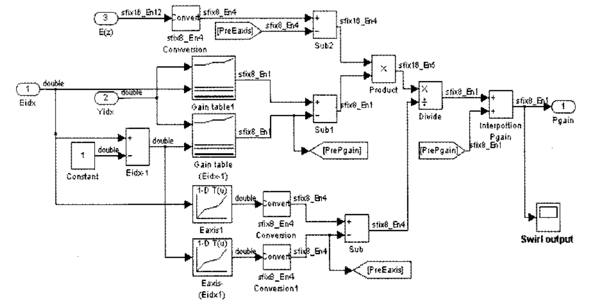
(b) FixPt GUI block



(c) SMC model with fixed-point



(d) Task-control block



(e) Interpolation Pgain block

Figure 6. Simulation SMC using fixed-point arithmetic.

Table 2. Gain look up table with fixed-point.

	E-axis	−44°	−6.6°	0°	6.6°	44°
Y-axis	Eidx Yidx	1	2	3	4	5
0°	1	1.5	6.5	24	9	2
22°	2	1.5	6.5	25	9	2
44°	3	1.5	6.5	27	9	2
66°	4	1.5	6.5	28	9	2
88°	5	1.5	6.5	31	9	2

floating-point arithmetic because fixed-point arithmetic leads the quantization errors and round-off errors. So gain look-up table needs to be modified through the simulation.

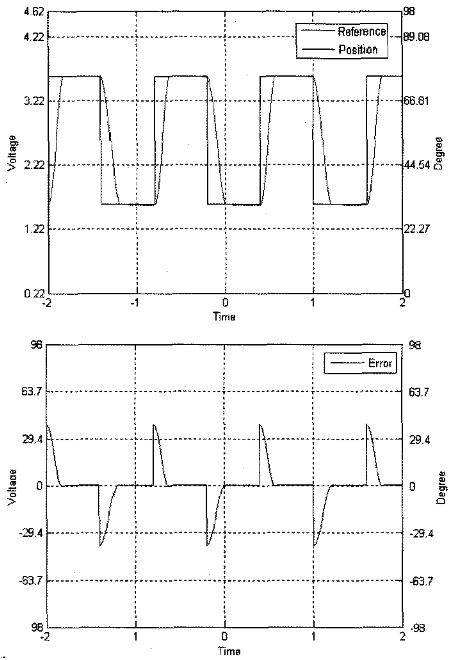
Table 2 is the gain look up table which describes conversion into fixed-point number. The gains are 8 bits consisting of 6 integer bits and 1 fraction bit. So decimal point is 1. The E-axis values are 4 integer bits and 4 fraction bits.

Figure 7 is the result of simulation of table 2. It has good performance with low steady state errors, fast response time and relatively low overshoot. Fixed-point conversion has the proper range and decimal point.

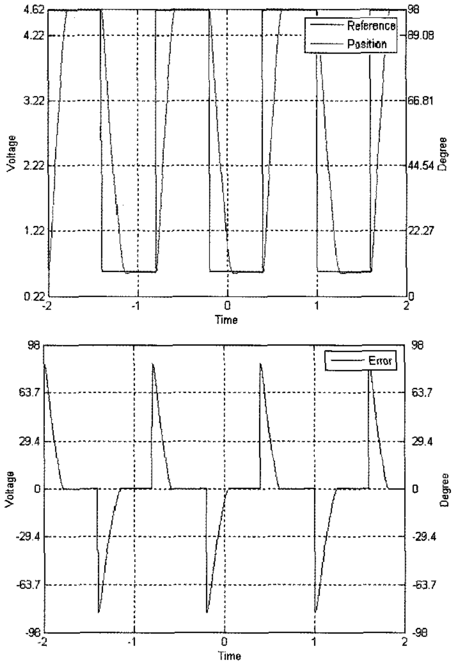
4. EXPERIMENTAL RESULT

The developed control algorithm is verified by the actual swirl control system experiment. For proving control algorithm, the experimental procedure must be same with that of the simulation, which verifies low steady state error, fast response time and relatively low overshoot according to the change of error and position.

Figure 8 is the structure of controller. Two inputs exist. One is reference input provided by a function generator instead of the output of ECU and the other is feedback input given by the potentiometer of swirl system. ADC



(a) Reference: 30 ~ 73°



(b) Reference: 10 ~ 97°

Figure 7. Controller performance using fixed-point.

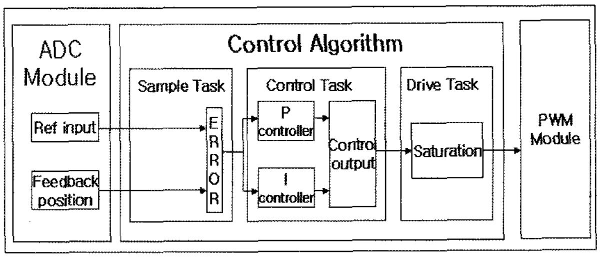


Figure 8. Controller block diagram.

Table 3. Gain look up table using experiment.

	E-axis	-44°	-6.6°	0°	6.6°	44°
Y-axis	Eidx Yidx	1	2	3	4	5
0°	1	2.5	5.5	22.5	7.5	4
22°	2	2.5	6.5	22.5	7.5	4
44°	3	2.5	6.5	23	7.5	4
66°	4	2.5	6.5	24	7.5	4
88°	5	2.5	6.5	26	7.5	4

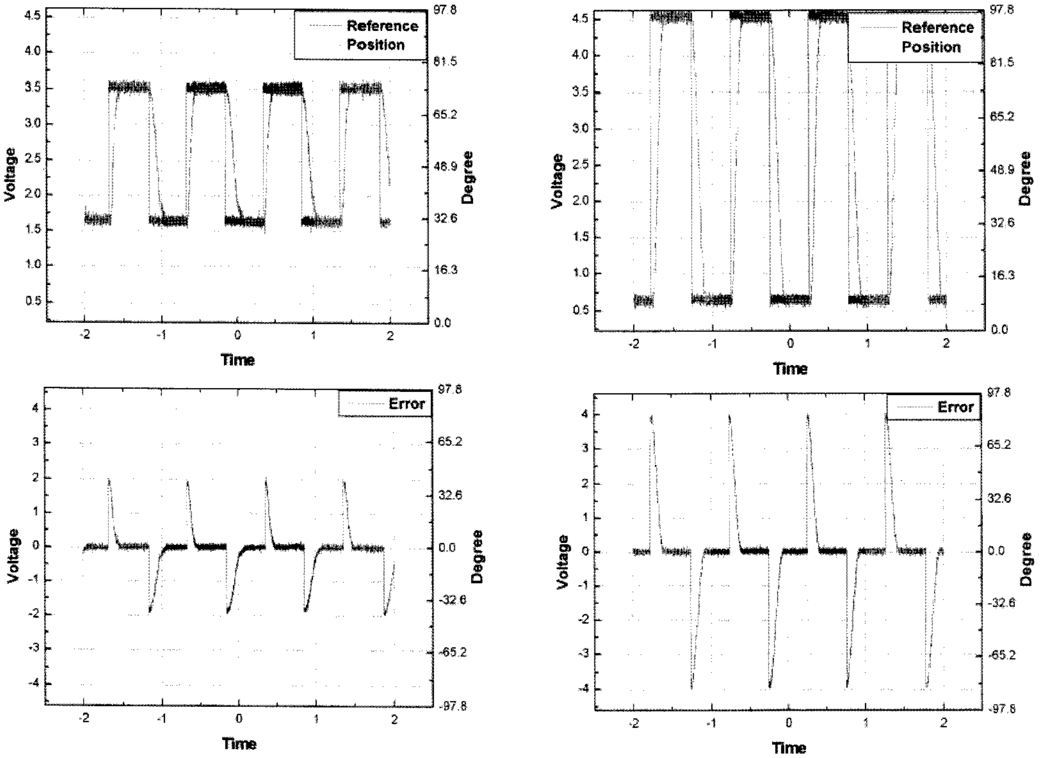
module converts analog input signal into digital signal. The control algorithm block operates a control algorithm instead of the SMC model in simulation. Task-sample calibrates the error that is result of subtraction of feedback input and reference input. Task-control operates proportional and integral control. Task-Drive calibrates PWM duty with task-control output. PWM module generates PWM signal that would be the input of motor drive IC. The output of motor drive IC actuates DC motor and controls angular position of swirl plate.

Table 3 is the gain look up table filled by experiments. It satisfies low steady state error, fast response time and relatively low overshoot. It is different from table 2 because of model error. And the integral gain is 30. It has good performance that compensates non-linear friction properly. Figure 9 shows the result of experiments according to the change of error and position.

5. CONCLUSION

Swirl system requires good performance, such as low steady state error, fast response time and relatively low overshoot because it largely affects on engine performance. Non-linear friction owing to mechanical structure makes the control of swirl system difficult. Although the same input is applied to swirl system, it produces different output according to the error and angular position.

Therefore the gain scheduling proportional controller compensates non-linear friction through gain look-up table that makes it possible to select the different gain according to error and angular position. It also offers fast response time and relatively low overshoot. And integral controller with anti wind-up that reduces the steady state errors was added to the control algorithm. Because the control algorithm is simple it can be implemented on low cost microcontroller, which has low computing speed and



(a) Reference: 30 ~ 73°

(b) Reference: 10~97°

Figure 9. Controller performance using experiments.

no-floating point, using fixed-point arithmetic.

The control algorithm of paper shows proper performance by compensating non-linear friction through the experiment.

REFERENCES

Charles, L. P. and Troy, H. N. (1995). *Digital Control System Analysis and Design.* 3rd Edn.. Prentice Hall. New Jersey.

Deur, J., Pavkovic, D., Peric, N. and Jansz, M. (2004). An electronic throttle control strategy including compensation of friction and limp-Home effects, industry applications. *IEEE Trans.* **40, 3**, 821–834.

Labrosse, J. J. (2000). *Embedded Systems Building Blocks. 2nd Edn.*. R&D Books. Kansas. USA.

Kang, J. J., Lee, W. T., Shin, K. C., Kim, J. S. and Chun, D. P. (2005). Development of algorithm for coulomb friction compensation in swirl system. *Spring Conf. Proc., Korean Society Automotive Engineers*, 431–436.

Pechlaner, A. and Björn, S. (2001). Electronic throttle control with contactless position sensor and smart power full-bridge, electronic engine controls. *SAE World Cong*. Detroit, Michigan March 5-8, 2001-01-0984.

William, F. R. and Leroy, D. S. (1996). *Engineering Mechanics Statics.* 2nd Edn.. Wiley, New York.

Yeo, H., Song, C. H., Kim, C. S. and Kim, H. S. (2004). Hardware in the loop simulation of hybrid vehicle for optimal engine operation by CVT ratio control. *Int. J. Automotive Technology* **5, 3**, 201–208.