DSpace VSB-TUO http://hdl.handle.net/10084/90303 19/07/2012

Michal KRUMNIKL, Petr OLIVKA

VŠB - Technical University of Ostrava

PWM Nonlinearity Reduction in Microstepping Unit Firmware

***Abstract**. Pulse Width Modulation (PWM) is a popular method for controlling analogue circuits with digital output of a microcontroller. This paper is focused on practical issues of using PWM in precise positioning systems. PWM is easy to grasp intuitively, but due to the motor characteristics, is not as easy to implement as one would expect. There is a distortion or nonlinear gain between the reference and real output voltage in the PWM system. In this paper, we will present a pure software solution to perform the PWM correction. We will introduce a novel algorithm capable of running on a small microcontroller. The algorithm is capable of performing all necessary computations for the PWM compensation without a need of feedback.*

Streszczenie.** Modulacja szerokości impulsów (PWM) jest popularną metodą regulacji sygnału analogowego z użyciem cyfrowego wyjścia mikrokontrolera. W artykule skupiono się na praktycznych aspektach wykorzystania PWM w systemach precyzyjnego przemieszczania. System PWM jest łatwy do zrozumienia, niestety z powodu cech silnika, jego zastosowanie nie jest tak łatwe, jakby mogło się wydawać. W systemach PWM pojawiają się zakłócenia oraz nielinearności zysku. W artykule opisano ściśle programowe rozwiązania korekcji PWM. Przedstawiono nowy algorytm nadający się dla małych mikrokontrolerów. Algorytm zawiera wszystkie potrzebne obliczenia dla kompensacji PWM bez konieczności sprzężenia zwrotnego. **Redukcja nielinearności PWM w oprogramowaniu sterownika mikrokrokowego.

Keywords: Step Motor, Microspepping Unit, PWM, PWM Nonlinearity Reduction, Microstepping Accuracy, Mikrocontroller.
Słowa kluczowe: proszę silnik krokowy, sterownik mikrokrokowy, PWM, kompensacja PWM, precyzja krokowania, mikrokontroler.

Introduction

Pulse Width Modulation is a popular method for controlling analog circuits with a digital output of a microcontroller. PWM is frequently used to regulate a current sent to a step motor. On most microcontrollers PWM is hardware based so once it is configured and enabled it does not require any additional CPU overhead. This paper is focused on practical issues of using PWM in precise positioning systems.

Precise positioning systems can be found in many industrial devices and robots. Accurate positioning is crucial, especially in configurations where several actuators run at the same time or are in different patterns of one continuous movement. Step motors are designed to maintain a fixed step for one turn, which makes it possible to achieve such accurate movements. In reality, there are a lot of applications which require even more sensitive control than fixed steps. Slow movements, fluent acceleration and deceleration or better accuracy in positioning are often required.

There is a way of achieving this by using special control technology, commonly known as microstepping. The principle, in short, can be described as a limitation of a current flow in motor winding. In this way, the step motor can be forced to operate in more positions than it was originally designed to. In practice, we usually use at most 8 or 16 microsteps between two mechanical positions. While theoretically it is possible to control the step motor continuously, in reality it is impossible. Internal friction will divide fluent motion into small steps.

PWM is easy to grasp intuitively, but due to the motor characteristics, is not as easy to implement as one would expect. PWM duty cycle is a measure of the fraction of the time that the device is in an active state. In our situation it is a period when the current flow through the winding and a magnetic field is formed. Generally, there is a distortion or nonlinear gain between the reference and real output voltage in the PWM system. The problem lies in the passive phase of the cycle, when no current flows. It is clear that PWM is not equivalent to the analogue signal. A nonlinear feedback is often used to reduce the deviations of a practical PWM output stage from ideal theoretical behaviour [1].

In this paper, we will present a pure software solution to perform the PWM correction. We do not require any type of feedback; we rely only on the calculations performed inside the firmware.

Our paper is organized as follows: In the opening section we will discuss the current research and development in the field of PWM drivers and microstepping units. Next, in Section 2 we will specify the requirements and our design goals. We will demonstrate our solution on a simple circuit, which has minimal impact on characteristics of a final power circuit. Next, we will provide more details on design ideas and derive main characteristics. In Section 3 we will focus on the firmware and accuracy of the proposed microstepping driver. Finally we will summarize the results and provide an outlook on further research.

Microstepping and PWM Nonlinearity

The effects of the PWM nonlinearity are widely studied and discussed. PWM is analysed in non-inverting buck-boost converters, where PWM discontinuity around buck/boost mode transition can result in substantial increases in output voltage ripple [2]. In a similar vein, nonlinearity in a PWM VSI (Voltage-Source Inverter) has to be estimated to compensate the time varying voltage distortion [4, 10]. In PWM rectifiers a nonlinear control scheme using state feedback is used to compensate the nonlinear switching effects of PWM [5]. PWM is also widely used to drive the winding of a bipolar stepper motor in a microstepping mode.

Microstepping was invented by Larry Durkos, a mechanical engineer of the American Monitor Corporation. Microstepping is actually a sine cosine driving in which the winding current approximates a sinusoidal AC waveform. This allows stopping and holding a position between two standard step positions and provides smoother operation in low speeds. PWM current control is usually combined with a nonlinear digital-to-analog converter which allows to control the motor current.

The basic principles of microstepping are described by Akdogan et al. [6]. This work is focused on simulation and demonstration of microstepping technique, providing the fundamental mathematical background for calculating the current flow in both phases.

A hardware implementation of a three phase driver for a two phase step motor is proposed by Yodsanti et al. [7]. The current generated by the PWM is measured and used as a feedback for real-time correction, thus achieving high precision. This is the classical approach when dealing with the PWM nonlinearity. Because the control is computational demanding, processors with floating point unit are needed for the compensation calculations. Houška et al. described the principles and general characteristics of a software controlled microstepping unit as well [8].

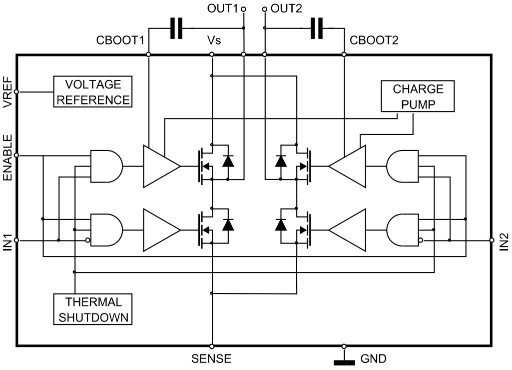

Fig.1. IC L6202 Block Diagram

Current designs are based on a general purpose microcontroller. Li and Zhou have used C8051F005 as a kernel, combined with L297/298 step motor driving chips[9]. IC L298 is a dual H-bridge in a single package, however without catch diodes and not so effective integrated bipolar transistors. Chen et al. adopted similar approach, using PIC18F2331 microcontroller to drive external power circuits using the PWM control signal [3]. The paper covers mainly the software part; hardware is not described in detail. Moreover, proposed processor is not suitable for USB communication.

In the following section we will analyse requirements needed for the compensation. Simple driving circuit will be used to demonstrate the benefits of our solution. The firmware is optimized for small and energy efficient units. We will show how to implement all necessary compensation functions into the microcontroller.

Microstepping Unit Driver

Before we can start to implement firmware functions, we need to specify the driver part it will control. Since we are focusing on embedded applications, we will design control unit for two phase step motors with power supply of 24V and maximum current 1A. Our goal is to create a simple solution to test our firmware. The solution must be tiny and light without the necessity of cooling. It should be constructed with minimum parts and integrated circuits.

In order to accomplish our goal, we have broken the work into several stages. In the first stage we will start with the design of the circuit itself. Next, we will focus on software compensation. By taking the advantage of an integrated approach, we have fully exploited the current technology. We have looked over the options and checked the majority of integrated circuits available on the market. We require an IC with integrated H-bridge. During the search for the optimal solution, we have eliminated ICs with half bridge, ICs without integrated catch diodes and ICs for high current applications. Then, by comparing the parameters of remaining ICs, we have selected the most suitable - ST Microelectronics L6202.

L6202 [11] is a circuit manufactured using the Multipower-BCD technology which combines isolated DMOS power transistors with CMOS and bipolar circuits on the same chip. The technical parameters are as follows:

- supply voltage up to 48V,
- current up to 1.5A,
- operating frequency up to 100kHz,
-
- TTL and CMOS compatible input of control logic.

Fully functional circuit can be assembled with only three small capacitors. Circuit has an internal logic power supply. Block diagram of L6202 integrated circuit is on the Fig.1.

Final power circuit diagram for two phase step motor is shown in Fig. 2.

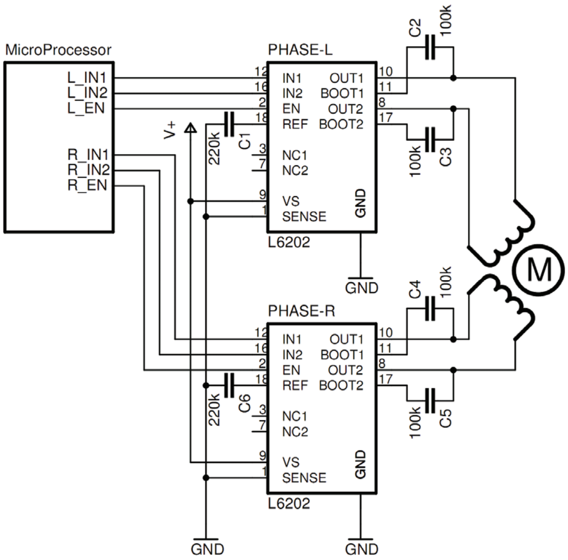

Fig. 2. Microstepping Unit Circuit Diagram

IC L6202 thermal analysis

Our observations indicate that the IC L6202 is the optimal driver for the tests of the microstepping unit firmware. The only parameter we have to carefully observe is the power dissipation. The unit is presented as a high efficient device, thus we expect operation without a cooler.

The electrical pulse from the integrated circuit can be divided into 4 parts - rising edge, open phase, falling edge and close phase. These phases are shown in Fig. 3. The dissipation energy can be calculated according the formulas available in the technical documentation.

The total dissipation energy is the sum of the following components: energy of the rising edge (E_{On}), falling edge (E_{Off}), energy during the period when two transistors are opened (E_{Open}) and quiescent energy (E_{Qui}).

$$E_{On} = R_{DS} \cdot I_L^{\,2} \cdot T_{On} \cdot 2/3 \tag{1}$$

$$E_{Off} = R_{DS} \cdot I_L^{\,2} \cdot T_{Off} \cdot 2/3 \tag{2}$$

$$E_{Open} = R_{DS} \cdot I_L^{\,2} \cdot T_{Open} \cdot 2 \tag{3}$$

$$E_{Qui} = V_S \cdot I_{Qui} \cdot T \tag{4}$$

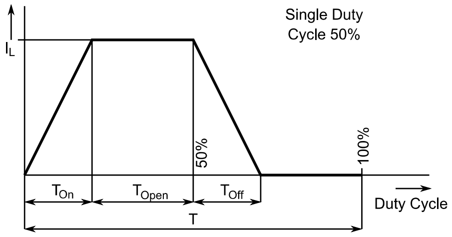

Fig. 3. Timing Diagram of Current in El. Circuit Over One PWM Period

The symbols in the formulas above have the following meaning: R_{DS} - internal resistance of an opened transistor, 0.3 at temperature of 25°C, V_S - supply voltage, expected maximum is 24V, I_{Qui} - quiescent supply current, typically 10mA, I_L - load current, expected maximum is 1A.

For practical application, we need to determine the worst case operating conditions. The worst situation that may occur is when the microstepping unit stops the step motor in a position, when one phase is fully opened. This means, one integrated circuit is continuously opened, providing the maximum current flow. In these situations, the maximum dissipation power can be calculated according the previous formulas as

$$P_{DIS} = V_S \cdot I_{Qui} + R_{DS} \cdot I_L^{\,2} \cdot 2 \tag{5}$$

The resistance of transistors R_{DS} depends on the chip temperature. Value 0.3 is a normalized value taken from the datasheet for the temperature of 25°C. The temperature dependency of R_{DS} is shown in the Fig.4. In our applications we do not expect operating temperatures exceeding 100°C. The coefficient   from the Fig. 4 is   = 1.4 for the temperature of 100°C. Maximum supply voltage V_S is 24V and quiescent current is 10 mA [11]. We can use the following modified formula to compute the maximum dissipation power of the integrated circuit:

$$P_{MAX} = V_S \cdot I_{Qui} + \alpha \cdot R_{DS} \cdot I_L^{\,2} \cdot 2 = 1.08W. \tag{6}$$

The maximum dissipation power, calculated as described by the formula 6 proves high effectivity of the construction. Having the value of calculated dissipation power, we can look at the datasheet for the required copper area [11]. The sufficient cooling for our application is achieved using the area of 2cm^2 of the copper on the circuit board, according to the Fig.5 taken from the technical documentation. This complies with our requirements for lightweight construction.

Accuracy of Microstepping

The goal of our design is to use the firmware of the microstepping unit for positioning. For this purpose, we need to check the precision of the microstepping unit. We have performed a series of angle measurements. For our experiments we have used two phase step motor SPA 42/100 with 100 steps per revolution. We have mounted a laser pointer on a motor axis and measured the microsteps using the beam projections on the wall at a distance of 4 meters. The control unit firmware was set to 8 microsteps.

The theoretical shift between the two microsteps is 0.45°per step. The theoretical positions are calculated in Tab. 1 (illustrated in Fig. 6) line 1. The measured values are in the same Tab. 1 at line 2. The table also contains medium values taken over ten experiments. All values in the table are in degrees.

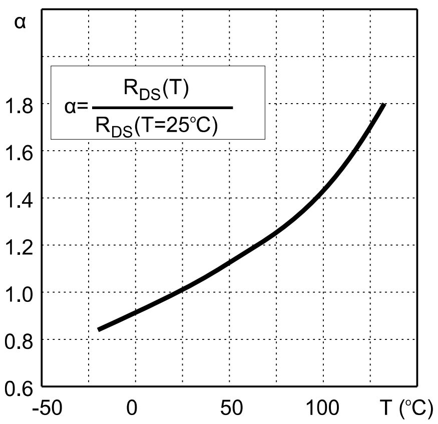

Fig. 4. Normalised R$_{DS}$ at 25°C vs. Temperature Typical Value

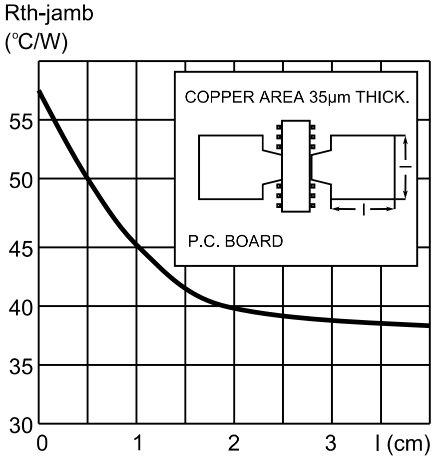

Fig. 5. Copper Area vs. Temperature Typical Value

The difference between the expected theoretical shift and real values was unexpectedly large. Therefore, we have checked the behaviour of other microstepping units as well. All small microstepping units controlled by the pulse wide modulation showed the similar inaccuracy. In continuous motion this inaccuracy is eliminate by the motor inertia. However, if we intend to use it for positioning purposes we must avoid all possible inaccuracies.

Table 1. PWM Microstepping - Measured Angle Position

step	0	1	2	3	4	5	6	7	8
1. Comp	0.00	0.45	0.90	1.35	1.80	2.25	2.70	3.15	3.60
2. PWM	0.00	0.26	0.69	1.27	1.77	2.28	2.88	3.33	3.60
3. OPWM	0.00	0.41	0.84	1.32	1.80	2.29	2.75	3.19	3.61

Optimal Current Feeding

In order to achieve high precision in positioning we need to improve the precision of current feeding. The I_1 and I_2 in the formula 7 are currents in the first and second stator windings. Phase shift is /2. The vector sum of currents in both stator windings must be a constant [6]:

$$\varphi_2 = \pi/2 - \varphi_1 \tag{7}$$

$$\sin\varphi_2 = \sin(\pi/2 - \varphi_1) = \cos\varphi_1 \tag{8}$$

$$\begin{aligned} I_{const} &= \sqrt{(I_{max}\sin\varphi_1)^2 + (I_{max}\sin\varphi 2)^2} \\ &= \sqrt{I_1^{\,2} + I_2^{\,2}} \end{aligned} \tag{9}$$

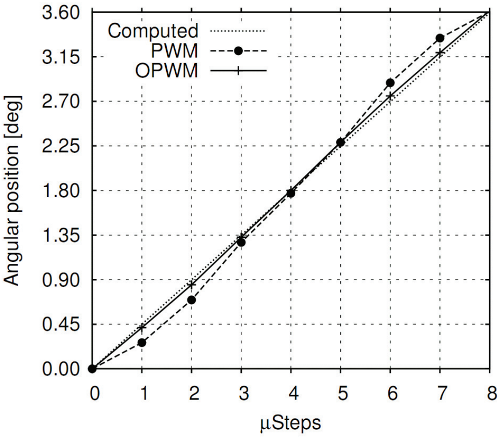

Fig. 6. Measured Angle Deviation

Table 2. Nonlinear Dependency of Current vs. Duty Cycle

step	0	1	2	3	4	5	6	7	8
Duty cyc. Ỳ	0	6.25	12.5	18.75	25.0	31.25	37.5	43.75	50.0
Current mA	0	3	6	11	18	25	34	43	53

step	9	10	11	12	13	14	15	16
Duty cyc. Ỳ	56.25	62.5	68.75	75.0	81.25	87.5	93.75	100.0
Current mA	63	72	84	96	115	138	157	171

First experiments showed that the direct use of PWM leads to irregular current feeding. In the further steps, we have focused on discovering the sources of this inaccuracy.

During the measurements, we have discovered non-linear dependency of current vs. pulse wide modulation. The main cause of this phenomenon is a nonlinear behaviour of the electrical circuit in step motor coils, mentioned also in [7]. The results of our measurements (for 16 PWM steps with increment 6.25%) are shown in the Tab.2.

The nonlinearity is clearly visible on the Fig. 7. This nonlinearity makes a computation of a corresponding duty cycle for a phase shift harder. It is clear that a straightforward use of the following formula is impossible:

$$DC = 100 \cdot \sin\varphi \tag{10}$$

Figure 7 is the proof of this observation; the real duty cycle has to be approximated from measured curve. The formula for duty cycle computation must be therefore more general as follows:

$$DC' = f(\sin\varphi) \tag{11}$$

The appropriate approximation is computed in the control software and stored in the conversion table. The conversion table can be easily recalculated for any step motor, providing thus higher accuracy.

The results obtained using the motor in our experimental configuration is in the Tab. 1 line 3. Measured accuracy is within 10\% deviation range from the expected value. The difference in accuracy between the standard PWM and optimized PWM (OPWM) can be found in the Tab. 3 and are illustrated in Fig.8. The deviations values are given in degrees. By taking into an account the internal friction of the step motor, the results are superior.

Table 3. PWM Microstepping - Angle Deviation

step	0	1	2	3	4	5	6	7	8
PWM	0.00	-0.19	-0.20	-0.07	-0.03	0.04	0.19	0.19	0.01
OPWM	0.00	-0.03	-0.05	-0.02	0.00	0.04	0.06	0.05	0.02

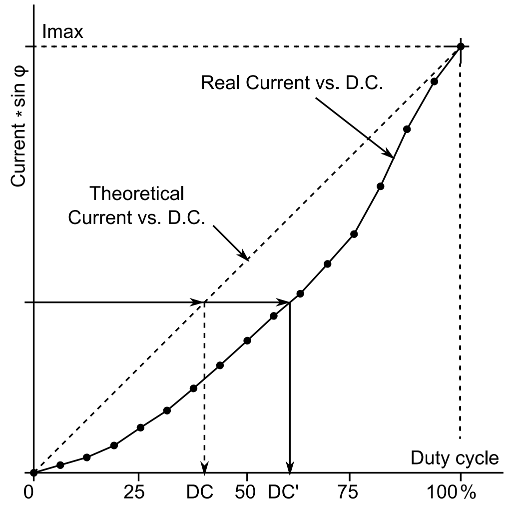

Fig. 7. Duty Cycle for Phase Shift

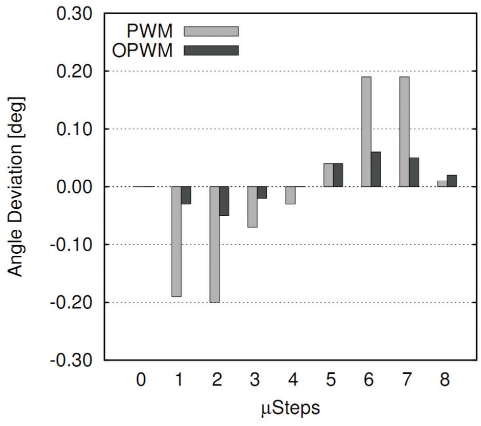

Fig. 8. Comparison of Angle Deviation of Standard PWM and Optimized PWM (OPWM)

Microcontroller Firmware

This section describes the kernel of the microstepping control program. We will focus on the main ideas and omit the non-relevant parts. The core of the microcontroller firmware is interrupt driven. Small parts of the source code are shown for the illustration in Lis. 1 and 2.

The basic operation of PWM is controlled by a microcontroller timer. An interrupt is generated each time the timer overflows. Frequency of the timer interrupt should be set according to the microcontroller oscillator. The interrupt handler is implemented in the function `interrupt timer( void )`. The source code for the interrupt handler is presented in Lis.1.

Duty cycle can be changed only at the begin of PWM period, when `pwm == 0`. When a new microstep position is activated, values of the duty cycle for L/R chip are copied to the variables `workL/R` and pins `L_IN1/2` and `R_IN1/2` are set to the correct polarization. Names and connections

```
#define MI_STEPS   8
#define MAX_STEPS (2*MI_STEPS)
#define PWM_MAX   16

unsigned char stepL, polL, pwmL, stepR,
                polR, pwmR;
unsigned char workL, workR, pwm;

interrupt timer( void ) {
  if ( pwm == 0 ) // pwm cycle start
  {
      if ( step_changed ) // new step?
      {
         // copy pwmL/R and set polarity
          step_changed = 0;
          workL = pwmL;
          L_IN1 = polL; L_IN2 = !polL;
          workR = pwmR;
          R_IN1 = polR; R_IN2 = !polR;
      }
      // Enable L/R Chip
      if ( workL ) L_EN = 1;
      if ( workR ) R_EN = 1;
  }
  // detect PWM Duty Cycle for L/R Chip
  if ( pwm == workL ) L_EN = 0;
  if ( pwm == workR ) R_EN = 0;

  pwm++;
  pwm &= PWM_MAX - 1;
}
```

Listing. 1. Interrupt Handler

of all pins are depicted on Fig. 2.

```
// One clockwise microstep
void ustep_up()
{
 stepL++;
 // when sine wave cross 0, change polarity
 if ( stepL == MAX_STEPS )
 {
   stepL = 0; polL ^= 1 ;
 }
 stepR++;

 // when sine wave cross 0, change polarity
 if ( stepR == MAX_STEPS )
 {
   stepR = 0; polR ^= 1;
 }
 // Aproximate optimal Duty Cycle for ustep
 pwmL = aprox_step2pwm( stepL );
 pwmR = aprox_step2pwm( stepR );
 step_changed = 1;
}

unsigned char aprox_step2pwm( unsigned char
step )
{ // Sine pulse is symmetrical
 if ( step >= MI_STEP )
    step = MAX_STEP - step;

// Compute optimal PWM Duty Cycle for
  current microstep
// DC' = tab_aprox(sin(step * (PI/MAX_STEPS)))
 ....
}
```

Listing. 2. Single Microstep with Correction

The duty cycle is controlled by the pins `L_EN` and `R_EN`. At the end of each cycle, when `pwm` counter overloads, its value is set back to zero.

Two functions `ustep_up()` and `ustep_down()` are suggested for the microstepping control in both direction - clockwise and anticlockwise. Valid microstep positions must be within the range `< 0,MAX\_STEPS-1>`. When a sine wave crosses zero, the polarity has to be set correctly and finally, correct duty cycle must be computed.

The source code of the function `aprox_step2pwm()` is shown in Lis. 2. The opposite direction should be implemented in the similar way. For approximation we have used formula 11 and data from Tab. 2. Internal functions used precalculated sine function with step /64.

It is important to mention that the calibration data in Tab. 2 must be measured for the specific step motor.

Conclusions

In this paper, we have proposed a novel algorithm for the microstepping control unit. The algorithm is a pure software solution providing the PWM correction. Implemented non-linearity correction increases the accuracy up to the level of motor internal friction. All necessary computations are performed inside the microcontroller.

In contrast to similar solutions, our design is without a need of the hardware feedback or current sensors. The firmware is not computationally demanding and does not require expensive digital signal processor.

Presented solution has been successfully deployed as a part of industrial application.

REFERENCES

[1] Craven, Peter; Hand, Larry; Attwood, Brian; Andersen, Jack, Digital PWM Amplifier Using Nonlinear Feedback and Predistortion, Audio Engineering Society Convention 128, 5, 2010

[2] Paul, D. Maksimovic, Analysis of PWM nonlinearity in non-inverting buck-boost power converters,"in Proc. IEEE PESC Conf. 2008, pp. 374 1-3747

[3] Chen Ge, Xv Fei-yun, Jia Min-ping, Hu Jian-zhong, Stepper motor microstepping controller based on PIC, Mechanical and Electrical Engineering Magazine, 2009-01, ISSN 1001-4551

[4] Hag-Wone Kim; et al; Nonlinearity estimation and compensation of PWM VSI for PMSM under resistance and flux linkage uncertainty, IEEE Transactions on Control Systems Technology, Volume 14, Issue 4, July 2006 Page(s):589–601

[5] Rioual P., Pouliquen H., Louis J. P.: Nonlinear control of PWM rectifier by state feedback linearization and exact PWM control, IEEE PESC Proceedings, June, 1994, pp.1095-1102.

[6] Akdogan, E., Topuz, V., Akbas, A., An education tool study on mechatronics: emulation of stepper motor driving systems by using a microcontroller based system interface, Mechatronics, 2004. ICM '04. Proceedings of the IEEE International Conference on, ISBN: 0-7803-8599-3

[7] Yodsanti, C., Konghirun, M., Implementation of two-axis highly accurate position control of space-vector PWM based microstepping bipolar drive, Electrical Machines and Systems, 2008. ICEMS 2008. International Conference, ISBN: 978-1-4244-3826-6

[8] Houška, P., Ondroušek, V., Věchet, S. and Březina, T., Control units for small electric drives with universal software interface, Recent Advances in Mechatronics 2007, Part 1, 185-189, DOI:10.1007/978-3-540-73956-2_37

[9] Li Zhi-qiang, Zhou Jie, Ren Sheng-jie, Control system of fine drive for step motor based on SCM, Mechanical and Electrical Engineering Magazine, 2007-07, ISSN 1001-4551

[10] A. M. Hava, Carrier based PWM-VSI drives in the overmodulation region, Ph.D. dissertation, Univ. Wisconsin, Madison, 1998.

[11] STMicroelectronics datasheet, http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/ CD00000089.pdf

***Authors**: Ing. Michal Krumnikl, Ing. Petr Olivka,Department of Computer Science,Faculty of Electrical Engineering and Computer Science,VŠB Technical University of Ostrava, tř. 17. listopadu, 708 33 Ostrava Poruba, Czech Republic, Email: michal.krumnikl@vsb.cz, petr.olivka@vsb.cz*