DIGITAL CONTROL SYSTEM SYNTHESIS FOR THE OWI-535 ROBOTIC ARM EDGE MANIPULATOR

Yassen Gorbounov[1], Stefan Petrov[2], Tihomir Dzhikov[3]

[1] *University of Mining and Geology "St. Ivan Rilski", E-mail y.gorbounov@mgu.bg*
[2] *University of Mining and Geology "St. Ivan Rilski", E-mail stivannet@gmail.com*
[3] *University of Mining and Geology "St. Ivan Rilski", E-mail itihomir.dzhikov@gmail.com*

ABSTRACT. An upgrade of the OWI-535 ROBOTIC ARM EDGE manipulator is discussed in the paper. It is done by building a multichannel PWM modulator to control the power circuits of the DC motors of the five coordinates of the manipulator. The modulator is built on a programmable logic device that allows easy configuration and scaling of the controller. Running parallel algorithms is inherent to this type of devices, which offers a significant advantage over conventional processors. A proposal is made to control the manipulator wirelessly over a mobile platform (smart phone or tablet) by using the Arduino open source platform as an intermediate controller that links the Bluetooth serial channel and the multichannel PWM modulator.

Keywords: Robotic manipulator, Arduino, Pulse Width Modulation (PWM), Programmable Logic Device (PLD)

СИНТЕЗ НА ЦИФРОВА СИСТЕМА ЗА УПРАВЛЕНИЕ НА МАНИПУЛАТОР OWI-535 ROBOTIC ARM EDGE

***Ясен Горбунов*[1], *Стефан Петров*[2], *Тихомир Джиков*[3]**
[1] *Минно-геоложки университет "Св. Иван Рилски", 1700 София, E-mail y.gorbounov@mgu.bg*
[2] *Минно-геоложки университет "Св. Иван Рилски", 1700 София, E-mail stivannet@gmail.com*
[3] *Минно-геоложки университет "Св. Иван Рилски", 1700 София, E-mail tihomir.dzhikov@gmail.com*

РЕЗЮМЕ. В статията е разгледана модернизация на манипулатора OWI-535 ROBOTIC ARM EDGE. Това е направено чрез изграждане на многоканален ШИМ модулатор за управление на силовите схеми на постояннотоковите двигатели на петте координати на манипулатора. Модулаторът е изграден на базата на програмируема логическа схема, която дава възможност за лесно конфигуриране и мащабиране на управлението. За тези схеми е присъща възможността за паралелно изпълнение на алгоритми, което е съществено предимство спрямо конвенционалните процесори. Предложена е възможност за създаване на безжично управление през мобилна платформа (смартфон или таблет) чрез използване на платформата с отворен код Arduino в качеството на междинен контролер, осъществяващ връзка между Bluetooth сериен канал и многоканалния ШИМ модулатор.

Ключови думи: Манипулатор, Ардуино, Широчинно-импулсна модулация (ШИМ), Програмируеми логически схеми

Introduction

The OWI-535 ROBOTIC ARM EDGE (OWI Inc., 2017) is a robotic hand with 5 degrees of freedom suitable for educational and training purposes (see in Fig. 1). The drive system for all coordinates is comprised of DC motors with gearboxes for which mechanical overload protection is provided. The control is carried out remotely on a wired connection, with the possibility of simply switching on or off the respective motor without changing the speed. The original schematic of the manipulator is depicted in Fig. 2. As is seen from the figure, the direction reversal is done with two-way switches. This limits the control options to merely manual control with no possibilities for any algorithmic control implementation or future extensibility. Furthermore, the motors exercise no torque at zero speed which means that one must rely on the capabilities of the mechanics alone.

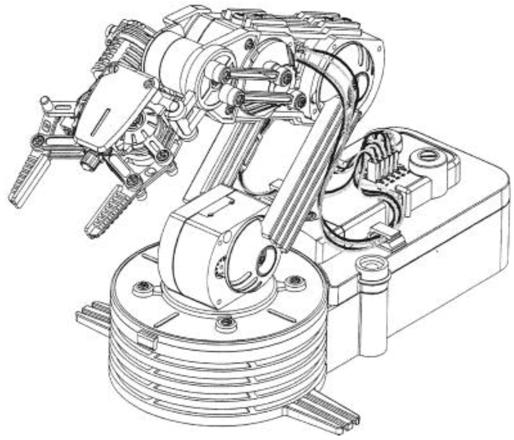

Fig. 1. OWI-535 Robotic Arm Edge Wired Controlled Arm Kit

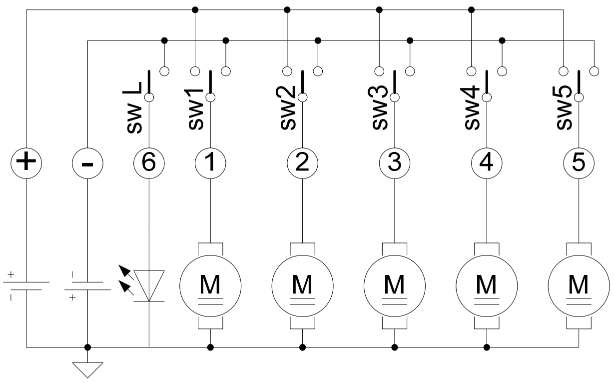

Fig. 2. OWI-535 Robotic Arm Edge original schematic

This paper discusses an upgrade of the drive system by implementing a PWM controller with dead time that is suitable for controlling full H-bridge power stages. This allows implementing both soft-chopping and hard-chopping switching strategies. This approach provides several advantages:

- Smooth control over a wide speed range;
- Achievement of a significant holding torque at zero speed;
- High efficiency coefficient with full protection of the power transistors with no shoot-through.

The PWM module works at the lowest control level. The fact that it is embedded in a programmable logic device allows for all the axes to be driven simultaneously, which means that it can be easily expanded over industrial grade manipulators that run real-time algorithms.

After introducing the H-bridge driver and the PWM module, a proposal is made in the paper for using the open-source Arduino platform acting as an intermediate controller that links a remote mobile device, such as a mobile phone or a tablet PC, with the low-level control logic via a wireless Bluetooth serial connection. This allows the user higher-level interface to be designed on such a platform as Android OS or similar. The final objective, however, is to enable the OWI-535 manipulator to interface with scientific products such as Matlab or its free equivalent – Scilab. This will make it possible to study various algorithms for trajectory control, interpolation, etc.

H-bridge motor drive

In Fig. 3 below, a full H-bridge is presented.

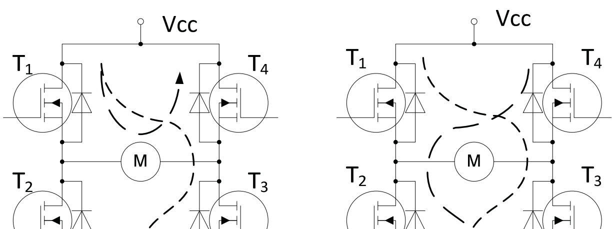

Fig. 3. Full H-bridge power stage built with N-channel MOSFETs

N-channel MOSFET transistors are used as power switches. They are controlled by voltage pulses and typically have internal resistance in the order of few milliohms when in on-state. This leads to very low active power dissipation. Since the motor windings are inductive in nature, a flyback diode is connected in parallel to the gate and source terminals. Two basic control strategies can be implemented with this circuit, namely: soft-chopping mode (Fig. 3, a) and hard-chopping mode (Fig. 3, b).

In soft-chopping (or non-symmetrical) mode, the transistor T1 is turned on for the entire phase excitation period while T2 and T4 are cut off. The diametrically opposed transistor T3 is controlled by the PWM pulses. When T3 is cutoff, a voltage with reverse polarity is created that attempts to maintain the value of the current flowing through the winding. There are two options: the current loop closes through T1 and the flyback diode of T4, or transistor T4 must be switched on to dissipate the stored energy. This method is not suitable for control at high speeds as the phase current diminishes relatively slowly and in some machines this can lead to a negative torque. In soft-chopping mode, the switching losses are lower and, hence, the energy efficiency is higher. This is good for the load in regards to the temperature.

In hard-chopping (or symmetrical) mode, all the transistors in the circuit are actively switching. In this mode, the switching losses are significant, which results in efficiency reduction. The increase in the operating frequency is limited as the switches have certain heat dissipation capabilities. This is a heavy operating mode in terms of temperature. On the other hand, this is the only way to achieve a static torque at zero speed. In this mode, both the speed and the direction depend on the PWM duty cycle, which is the same for each pair of transistors, T1-T3 or T2-T4. In this mode, the motor will rotate in the positive direction when the duty cycle is more that 50% and it will rotate in the negative direction when it is less than 50%. With a duty cycle of 50%, the motor is stopped despite the fact that it is fully loaded electrically. The latter guarantees the presence of a static torque at zero speed. One main peculiarity of this circuit is that a shoot-through can emerge if the transistors T1 and T2 or T3 and T4 remain switched on at the same time. Such a situation can occur if T2 is turned on before T1 cuts off and this can happen because of the recovery time of each transistor which prevents it to be cut off instantaneously. To ensure proper operation, the bridge shoot through should be avoided. This necessitates the introduction of the so called "interlock delay time" or "dead time" into the control scheme which represents a short period that ensures some time for the corresponding power switch to be fully turned off. The dead time calculation will not be discussed here but it should be mentioned that it is typically in the order of 0.3 to 5us (Selva, 2014; Infineon Technologies AG, 2007).

Pulse width modulator

The Pulse Width Modulation (PWM) is a technique for obtaining analog voltages by digital means. It is commonly used in power switching circuits like switched power supplies or motor inverters. For producing PWM, digital control is used to create a square wave which is a signal switched between on and off. This is usually done using a triangle-shaped voltage and a comparator. Unfortunately, this very simple setup is not suitable for driving H-bridge topologies. Such circuits need the introduction of some current-less time period, known as the dead time. A two-way PWM module with configurable dead time is discussed below. It is written in Verilog HDL and implemented in a programmable logic device of the CPLD type. The block diagram of the module is given in Fig. 4.

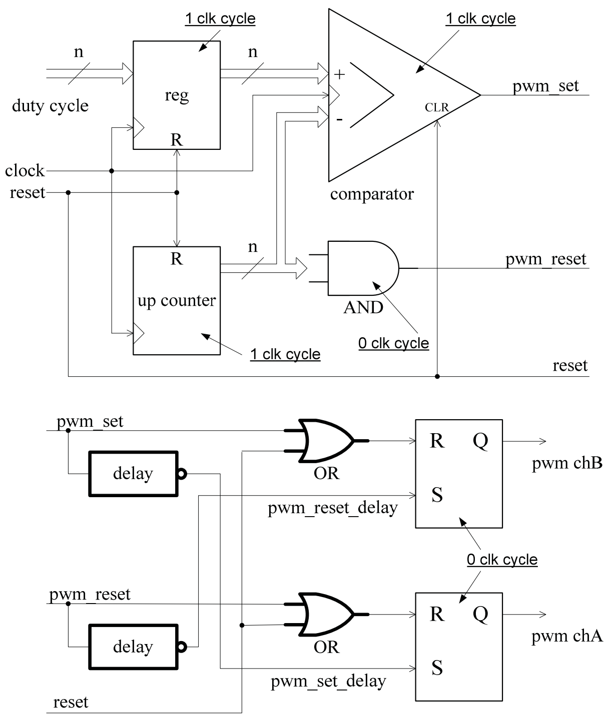

Fig. 4. Block diagram of the PWM module with dead time control capabilities

The operating principle of the circuit is based on the use of a counter and a comparator. When the reset signal is low (log. 0), the input register "reg", the counter, the comparator, and finally the two RS flip-flops at the output are all zeroed. If the reset goes high (log. 1), then at the first rising edge of the clock, the duty cycle assignment is stored in the input register "reg" and the up-counter increases by one. When the counter value reaches the duty cycle value, the comparator sets its output to log. 1 (rises the "pwm_set") thus resetting the output of channel B (see Fig. 5). At the same time, this signal starts up a monostable multivibrator (the upper "delay" block in the figure) which outputs a pulse with predefined length. Its output is negated and serves to set the output of channel A (see Fig. 5). The counter counts up and when it reaches the maximum value (all bits in log. 1 state; it saturates), the output of the AND gate ("pwm_reset") resets the output of channel A. At the same time, this pulse starts up a monostable multivibrator (the lower "delay" block in the figure) whose output serves to set the output of channel B. On the next rising edge of the clock, the counter overflows and starts counting again from 0. Then the whole algorithm repeats.

Fig. 4 shows the clock cycles that are consumed by each element. As is seen, all clocked modules consume 1 clock cycle, i.e. their output becomes valid after 1 clock period. All the asynchronous modules switch their outputs immediately.

The bold blocks in the figure represent the logic for implementing the dead time delay. The length of the delay pulse can be configured by setting a parameter in the "delay" block which technically leads to adding serially a flip-flop inside this block. That means that each delay unit corresponds to one clock cycle. The introduction of such a delay module allows for a precise setting of the dead time up to the resolution of the clock. Naturally, it can be assumed that the duty cycle of this PWM device is limited at both ends by the duration of the dead time. Therefore, the duty cycle should not be less than the dead time. If this condition is violated, the two channels will overlap and will be active at the same time, which is a prerequisite for a shoot-through and must be avoided. It is obvious that the higher the frequency and the resolution of the PWM are, the less the impact of this drawback is. In practice, this limitation is negligible as it falls outside of the useful operating mode.

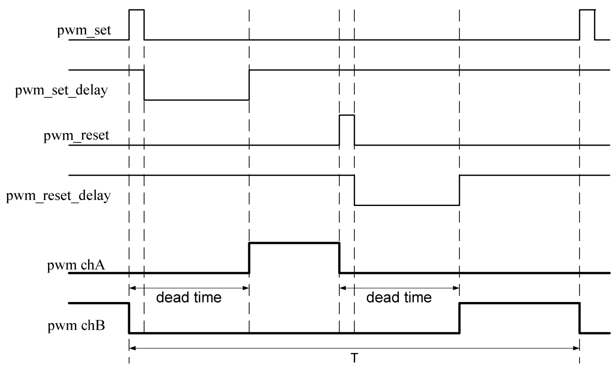

Fig. 5. Waveform that clarifies the principle of operation of the PWM

Fig. 6 and Fig. 7 show the experimentally obtained waveforms of an 8-bit PWM with clock frequency of 100 kHz. That means that the time resolution is 10us and the frequency of the PWM is about 390Hz. The usage of a CPLD permits the resolution to be configured to almost any value such as 9, 13 or 24 bits if needed. Also, the clock frequency can be augmented in the order of tens to thousands MHz. In the figures, the dead time, which is 30us for the sake of the experiment, can easily be seen.

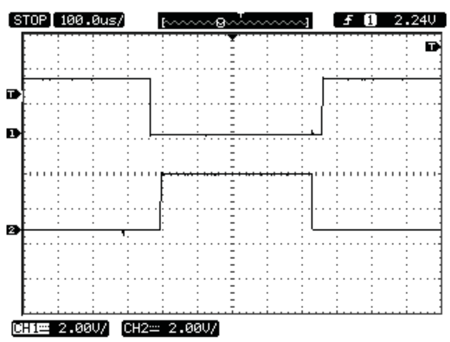

Fig. 6. Output of channels A and B for an 8-bit PWM with duty cycle of 0xD0 at 100 kHz clock

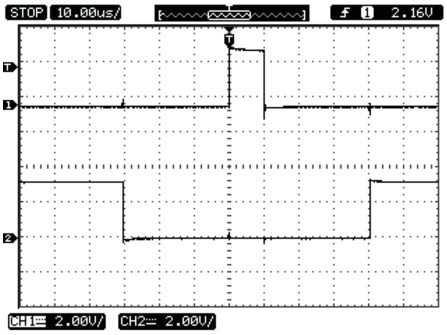

Fig. 7. Output of channels A and B for an 8-bit PWM with duty cycle of 0x02 at 100 kHz clock

The elaborated PWM module can be easily modified to support multiple parallel running channels. The setup is depicted in Fig. 8. As is seen, the up-counter and the AND gate are common to all channels, so they are instantiated once for the entire device. The duty cycle register, the comparator, and the delay logic are repeated for every channel. The number of channels depends on the available resources on the selected chip and does not affect the performance of the module as the CPLD makes them running in parallel. For the OWI-535, the PWM consists of 5 channels – one channel per coordinate. The usage of CPLD device resources increments linearly with the increase of the channel count and this makes the design flow quite predictable.

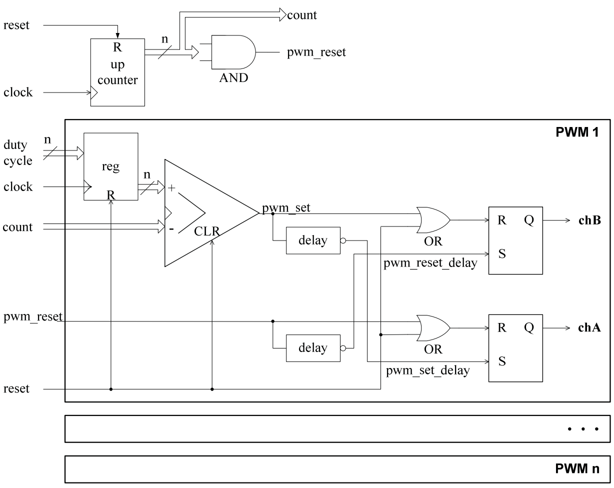

Fig. 8. Multichannel PWM configuration

Arduino intermediate controller

The PWM module discussed so far serves as a low-level controller for driving the motor. For the OWI-535 manipulator to be controlled, a higher-level device is required. The proposal here is to use the Arduino open source platform as an intermediate controller that makes a bridge between the PWM module and the user interface, or to use it as a standalone controller that generates the assignments for each axis. A block diagram of such a configuration is depicted in Fig. 9.

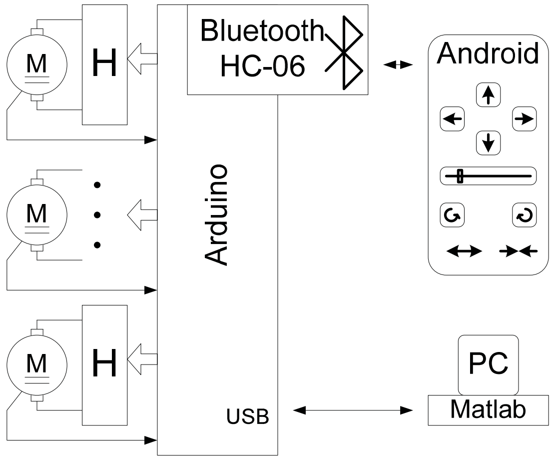

Fig. 9. Remote control of the OWI-535

Two options are shown. First, it is the Bluetooth serial connection. The main function of the Arduino is to translate the serially received commands to duty cycle assignments for each axis of the manipulator. Besides, the Arduino will check for boundary conditions for the assignments. This configuration will permit the manipulator to be controlled remotely. Second, it can be the interface between the OWI-535 ROBOTIC ARM EDGE manipulator and the commercial Matlab or the open source Scilab mathematical tool. This will give opportunity for studying algorithms and applying them to a real robot. And this is the final goal of this project.

The precise control of any device requires feedback information, so this is another function of the Arduino. A quadrature encoder can be used to obtain positional feedback. Another option is to build a sensorless control system by using a current sensor such as a Hall-effect sensor or a current transformer. It is convenient to implement maximum current control for limiting the force and protecting the motors. Furthermore, building a current feedback will provide information about the loading on each axis.

Future work

The results reported in this paper open possibilities for future work. This includes interfacing the OWI-535 ROBOTIC ARM EDGE manipulator with the Matlab scientific software. If a bigger programmable logic device (FPGA) is to be used, then an embedded microprocessor, such as Microblaze Soft Processor Core (Xilinx Inc., 2017) or Nios II (Intel Corp., 2017), can be implemented into the chip making the whole controller a standalone intelligent module. Also, SoC devices exist which incorporate FPGA fabric together with a single or dual core ARM MCU in a single chip.

Conclusions

An upgrade of the OWI-535 ROBOTIC ARM EDGE manipulator has been discussed in the paper. As a core component, a pulse width modulator module with embedded dead time control was described. The functioning of this module has been verified in practice on a XC2C256

CoolRunner CPLD from Xilinx. Two switching strategies have been evaluated: the soft-chopping and hard-chopping modes. Although being unfavorable for the motor because of the greater power losses, the hard-chopping mode proved to be more suitable for precision control of positioning devices since it guarantees significant static torque at zero speed and smooth control over a wider speed range.

The authors hope that the synthesized digital control system can be applied in several course units in the Department of Automation of Mining Production at the University of Mining and Geology "St. Ivan Rilski" in Sofia.

Acknowledgements

This work is supported by contract No MEMF-147/29.03.2017, University of Mining and Geology "St. Ivan Rilski", Sofia.

References

Gorbounov, Y. Fundamentals of digital design (in Bulgarian), University of Mining and Geology "St. Ivan Rilski", Sofia, 2016, ISBN 978-954-353-307-7

Gorbounov, Y. Application of programmable logic devices in electric drives (in Bulgarian), University of Mining and Geology "St. Ivan Rilski", Sofia, 2016, ISBN 978-954-353-306-0

Infineon Technologies AG, How to calculate and minimize the dead time requirement for IGBTs properly, AN2007-04, Application Note, V1.0, 2007

Intel Corp., Nios II Processor, USA, 2017, https://www.altera.com/products/processors/overview.html (Accessed July, 2017)

OWI Inc. dba: RobotiKits Direct, 17141 Kingsview Avenue, Carson, CA 90746, USA, http://www.owirobot.com/robotic-arm-edge-1/ (Accessed July, 2017)

Pavlitov, K. Programmable logic in electromechanics (in Bulgarian), Technical University of Sofia, ISBN 978-954-438-645-0, 2007

Selva R., V. Karthick, D. Arun. A Review on Dead-Time Effects in PWM Inverters and Various Elimination Techniques, School of Electrical and Electronics Engineering, SASTRA University, Thanjavur, Tamil Nadu, International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 1, January 2014

Xilinx Inc., MicroBlaze Soft Processor Core, USA, 2017, https://www.xilinx.com/products/design-tools/microblaze.html (Accessed July, 2017)

The article is reviewed by Assoc. Prof. Dr. Diana Decheva and Assoc. Prof. Dr. Zdravko Iliev.