AGV Robot Control by using FPGA

Alireza rezaee

Amirkabir University of technology
IRAN

Abstract: In this paper, the robot with three wheels is implement and its motion is controlled . two of robot's wheels are activated by DC motors. The main PID controllers are implemented with a Field Programmable Gate Array(FPGA) for independent robot movement .
a FPGA controller communicate with a PC with PCI card(PLX9052) so that the movement of wheels can be displayed on the monitor through PCI card communication. Experimental results show that good position tracking can be achieved .In addition, on linear digital PID controller is also implemented on AT mega128L(AVR micro controller) to compare with FPGA.

Keywords: Motion controller,FPGA,ATmega128L,Robot

1. Introduction

The emergence of reconfigurable Field Programmable Gate Arrays (FPGA) has given rise to a new platform of complete Mobile robot control system. With FPGA devices, the designer may tailor the design to fit the requirement of control system functions for a mobile robot. General-purpose computer can provide acceptable performance for mobile robot control when the tasks are not too complex .Because the single processor system cannot guarantee real-time response for any task when the complexity of tasks have been increased[1].A FPGA-based control system is designed to solve the platform of parallel tasks achieving control which Occurs on single processor machine. The parallel scheme has been applied toward

Figure1.Previous robot

this design without any problem of performance and the availability of I/O channels.

A FPGA-based robot control chip improves upon the single processor computer in following areas:

- Scalability of the control system, FPGA chips can integrate and still perform the pervious defined tasks efficiently, especially when more tasks achieving are required.

- Increase the availability of I/O channels.
- Directly map the logical designed to the computing elements in FPGA devices.
- Support parallel design scheme which leads itself to guarantee the control response time, real-time.
- Low power consumption compared to the single processor computer or even the micro-controller.

In our previous researches, we have built the robot hand system that uses 80196KC micro controllers and ISA card .It requires a serial communication box to communicate micro controllers with a PC or need ISA CARD. The overall system is shown in figure 1 and the whole system appears to be bulky [1].

Here in this paper ,instead of using those massive hardwares we designed a single FPGA chip to perform the same motion control of robot wheels. We developed a general purposed motion control of robot using a field programmable gate array(FPGA).The FPGA is known as a programmable hardware device that a user can design his/her algorithms and download it to a single chip. By using the high modifiable capacity of an FPGA, the additional hardware such as an encoder counter and a PWM generator, can also be implemented in a single FPGA device. As a result, the total controller size and power dissipation are effectively reduced by using an FPGA motion controller. Experimental results show that the robot follows the desired speed trajectories.

2.OVERALL SYSTEM

The newly designed overall system block diagram of robot control is shown in Figure 2.It consists of robot ,an FPGA controller, PCI Card ,a motor driver , ,and a PC.

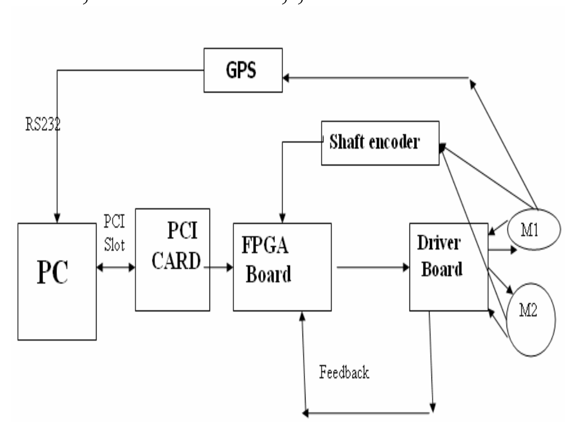

Figure2.overall structure of robot

Different from the previous model shown in figure 1, an FPGA replaces a communication box as well as several MCUs or ISA card.
The PC communicates PCI card through a PCI slot. The PCI car d communicates with a FPGA chip to transfer desired speed and controller gains at each sampling time. The FPGA chip programmed for PID controller generates PWM signals to motor drivers. Then each motor is actuated and each motor's encoder measurement is transferred to FPGA chip for making positional errors.

We also developed a PC GUI program for an efficient interface control between a motor and a PC. The GUI has several functions such as setting of PID control parameters, downloading of trajectory, displaying the control states, and so on. A user can specify PID controller gains through GUI program in PC. Those gains are transmitted though a PCI communication to a PCI card with Wind river function .And then it sends them to an FPGA.

The interfacing GUI program helps users to manipulate controller's gains and to observe the response of each motor. Figure 3 & 4 shows the GUI window that has many functions.

Each part has the following function:

1-motion control part (start, stop)
2-PID gain selection
3-Trajectory planner
4- Motors Selection
5-diagniose of PCI card

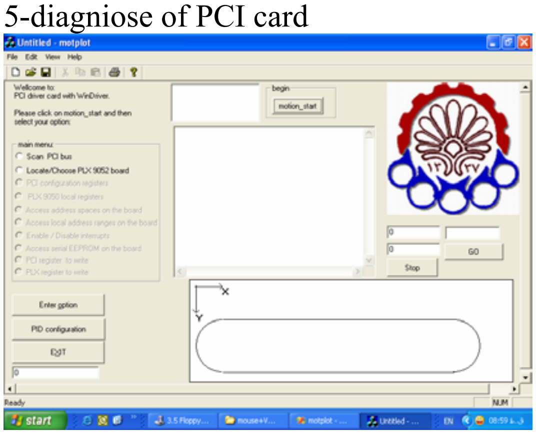

Figure.3 GUI on PC

3-PID CONTROLLER ON FPGA

3.1Overall structure

PID control method is very simple and important especially in robot control applications. The PID controllers are implemented in a FPGA chip having (Spartan II) 150,000 gates made from Xilinx company.

The design of PID motion controller is programmed by ISE6.2.

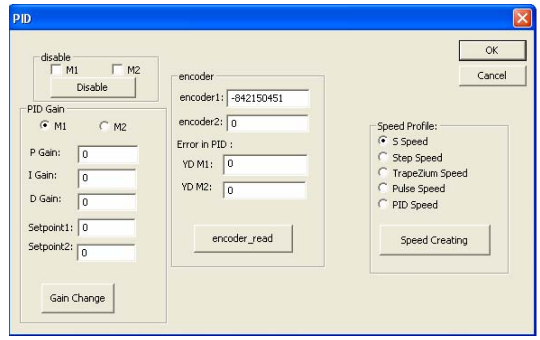

Figure 4.PID GUI

Figure 5 shows the schematic design of the controller .The FPGA based PID controller consists of several blocks such as a communication block, an encoder counter block, a PID calculation block, and a PWM generation block.

Table 1 lists functions of each pin show in figure 5.

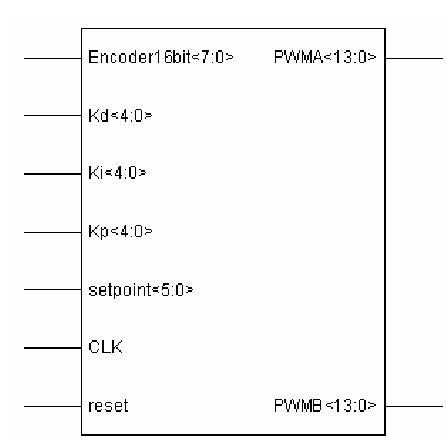

Figure5 .The design of motion controller on ISE

3.2 The communication block

It receives PID gains and desired trajectory, and transfers encoder data from the controller to an FPGA whenever they are needed. Data write from the PC to FPGA to the motion controller included PID gains, enable/disable, and the motion controller reset. Table 2 shows commands and their data.

Data read from the motion controller to an MCU is to read encoder data. Since the size of encoder data is 16 bit ,it should be read as high and low bytes separately.

	Pin name	functions
Input	M1_CHA	Motor 1 encoder phase A
	M1_CHB	Motor 1 encoder phase B
	M2_CHA	Motor 2 encoder phase A
	M2_CHB	Motor 2 encoder phase B
	main_clk	FPGA system clock
	Cs	Chip select
	Rd	Read
	Wr	Write
	c_d	Command, data
	data_HL	High, low byte
Output	M1_motor_in1	Motor 1 output 1
	M1_motor_in2	Motor 1 output 2
	M2_motor_in1	Motor 2 output 1
	M2_motor_in2	Motor 2 output 2
	enable1	Motor 1 enable
	enable2	Motor 2 enable
Bidirection	data[7..0]	8bits data bus

Table1.The Pin name and its function

Command	Data	Command	Data
Motor 1 Kp	0x01	Motor 2 Kp	0x11
Motor 1 Ki	0x02	Motor 2 Ki	0x12
Motor 1 Kd	0x03	Motor 2 Kd	0x13
Motor 1 Yd	0x04	Motor 2 Yd	0x14
Motor 1 enable	0x05	Motor 2 enable	0x15
Motor 1 disable	0x06	Motor 2 disable	0x16
Read encoder Motor 1	0x07	Read encoder Motor 2	0x17
FPGA reset	0x08		

Table2.Communication command

3.3 Motor control block

Figure 6 shows the motor control block. It consists of encoder counter, PID controller, and PWM generator .The clock synchronizes the process. It generates the trigger signal at each 1KHZ sampling time and sends it to encoder counter block and PID controller block to synchronize the process of encoder counter and PID controller.

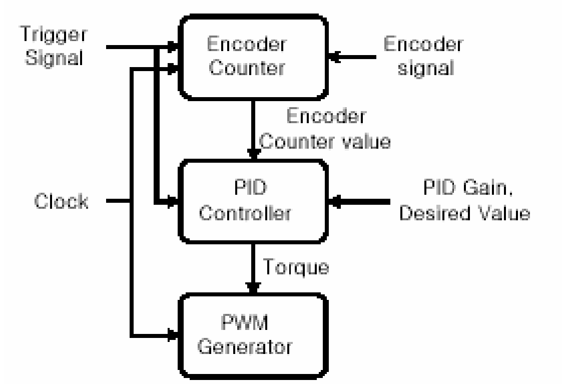

Figure 6. The structure of motor control block

Figure7 shows the designed motor control symbol. It takes PID gains, a clock, a trigger signal, encoder signals, a reset signal as input and PWM signal and encoder counter values as output.

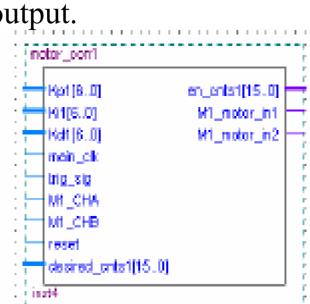

Figure 7. The designed symbol of motor control block

3.3.1 Encoder Counter Block

It counts and determines the direction of motor rotation from encoder signal. The block diagram is shown in figure 8.

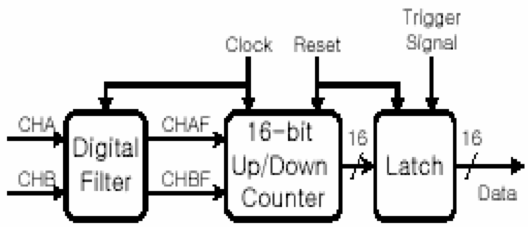

Figure 8. The structure of encoder counter

Different in phase A and B determines the direction of rotation. Noise from a mechanical system can be filtered out by digital filter as shown in figure 8. Every trigger signal enables to generate counter values.16 bit of counter limits the range of the movements.

3.3.2 PID controller block

It received encoder data from the encoder counter block and compares them with desired values to generate positional errors and then generate PID controllers. PID control equations are

$$\tau(n) = K_p e(n) + K_i s(n) + K_d\{e(n) - e(n-1)\} \qquad (1)$$

$$s(n) = \begin{cases} s_i & \sum e(n) > s_i \\ \sum e(n) & -s_i \le \sum e(n) \le s_i \\ -s_i & \sum e(n) < -s_i \end{cases} \qquad (2)$$

$\tau(n)$ Is control input, $e(n)$ is error

K_p, K_i, K_d Are PID gains, S_I is a threshold value.

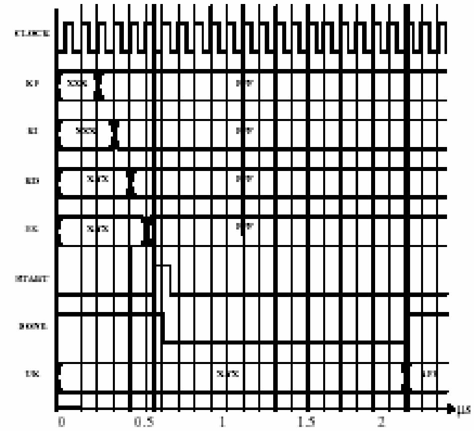

Figure 9.Timing diagram of the simulation of PID

4. AVR microcontroller code

The main PID controller routine was designed to be fairly general purpose and hence modular. Whilst here it is used to control a DC motor, it could be re-deployed to other situations where some parameter has to be controlled to a set value under varying conditions. The actual control software is located in a single function and its major inputs and output are held in a structure. Although it was designed originally for a specific job it is really only intended as an example of the basic techniques involved and to allow those with no control system knowledge to experiment with a simple PID system.

The routines that gather the inputs and process the output are kept in separate functions in another module. The code vision compiler was used .

For simplicity and to allow the easy modification of the important PID gain parameters, the AT mega 10-bit analog to digital converter was used to derive 10-bit resolution inputs from simple trimmer potentiometers. The PWM used to drive the motor was chosen as 10-bits so that motor speed can be defined to approximately 0.1%, sufficient for most practical application. The ATmega128 high resolution dedicated PWM unit could have been used to increase this to a supersonic 19.5kHz but to allow easy porting to other AT mega variants this route was not taken. Fortunately the use of a 10 bit resolution on the inputs and output makes some of the arithmetic easier!

The Atmega128 IO pins are allocated as per:

P2.0 optical chopper encoder input

P2.1 channel 2 - PWM drive output

P5.0 analog channel 0 - set point input

P5.1 analog channel 1 - derivative gain input
P5.2 analog channel1 proportional gain input

P5.3 analog channel 1 - integral gain

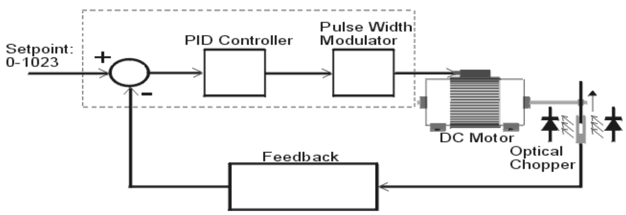

figure10. Implementation OF PID on AVRAtmega28

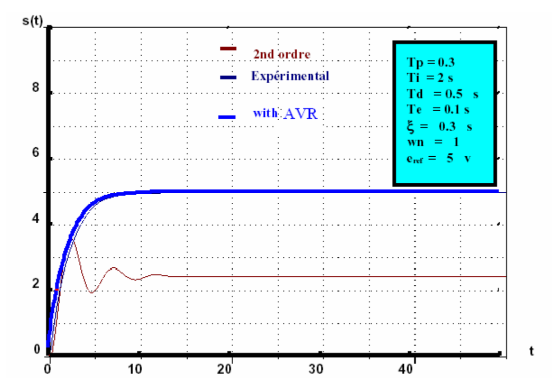

Figure 11.compare of PID on FPGA and AVR

5. CONCLUSIONS

In this paper, we have designed a directly driven robot with three wheels. The PID controller for two wheels is implemented by an FPGA. Control using an FPGA was successful and join movement are displayed on GUI.
Implementing controller with FPGA turned out to be very effective since the size of the whole system was remarkably reduced and it was cost effective as well.

REFFERENCES

[1]Jingo Kim, Sun Su Kim & Saul Jung ,"Implementation and Motion control of Linkage Bar Fingers",PP.2091-2094,ICCAS 2002.

[2]T.Aoyama,Qianyi Wang,R.Suematsu,R.Shimizu,U .Nagashima,"Learning algorithms for neural network in FPGA".Proceedins of the 2002 International Joint Conference on Neural Networks,vol.1,pp.1007-1012,2002.

[3]M.Krips,TLammert,A.Kummert."FPGA implementation of neural network for real-time hand traking system"Proceedings of the First IEEE Internatinal Workshop on Electronic Design,Test and Applications,pp.313-317,2002.

[4]M.Cristea,J.Khor,M.McCormick;"FPGA fuuzy logic controller for variable speed generators",Proceedings of the 2001 IEEE International Conference on Control Application,pp301-304,2001.

[5]Abdelkrim k.Oudjida et al,"A reconfigurable counter controller for digital motion control application",Microelectronics,Journal,vol.28,no.6-7,1997.

[6]F.Thomas et al,"Design and implementation of a wheel speed measurement circuit using field programmable gate arrays in a

spacecraft" Microprocessors and Microsystems,pp.553-560,1999.
[7]S.N oh et al,"Design of a bipid robot using DSP and FPGA",Proceeding 2002 FIRA Robot World congress,698-701.
[8]A.Kongmunvattana ,P chongstivatana,"A FPGA-Based behavioral Control system for a mobile Robot ",The 1998 IEEE Asia –Pacific Conference on Circuits and Systems,pp.759-762,24-27,Nov 1998.
[9]Sung-Su kim and Seul Jung,"Implementation of an intelligent controller with a DSP and a FPGA for nonlinear system" to appear in ICCAS2003.