A&QT-R 2002 (THETA 13)
International Conference on Automation, Quality and Testing, Robotics
May 23-25, 2002, Cluj-Napoca, Romania

P. W. M., SOFTWARE PID AND COMMAND TRANSMISSION, IMPLEMENTED IN C++, FOR INDUCTION MACHINE

Author : conf. univ. Dr. eng. Valentin Casavela,***University of Petrosani, department Hunedoara, Romania,e-mail:*** casavela@upet.ro***,phone: 054 54 16 81***

Abstract-This paper presents an interesting bipolar transistors inverter scheme, designed and built, and an isolated bipolar transistors (I.G.B.T) inverter scheme, designed also by the author . There, an absolute electrical isolation between the inverter itself and the I.B.M. P.C parallel output port may be seen, isolation achieved with some optocouplers.
The I.B.M. P.C parallel output port is used for to command the inverter, which is very simple, because the regulator, a proportional integral derivation (P.I.D.) one, is virtual, being included in C++ program. An acquisition card reads the incremental speed transducer and ties into the PC2 (mouse), one of the I.B.M. P.C serial input port.
C++ offers a very friendly graphic customer interface; this being actually the computer display. This program asks, for the beginning, the parameters of the induction motor and, if the motor of the stand is used, its parameters are listed. If there is another motor, the customer must introduce those parameters. Some figures, copied from the display, during the program was running, are included into the paper. That program also offers an automatic machine circle, which may be used for to command a robot, for example, circle that may be modified after the customer's will.

Key words: inverter fed induction machine, software pid.

I. INTRODUCTION

Induction motors are inexpensive and reliable. Controlled by an appropriate μP, a PWM inverter powered by a single d.c. supply, can generate waveforms that appear to such motors as the 3-phase, variable-voltage, variable-frequency sine waves PWM they need for speed control. Because the motor's speed and acceleration depend on amplitude as well as frequency, the inverter must produce sine waves of variable voltage and frequency. Using Stator Field Oriented Speed Regulation, which obtains a constant maximum torque, irrespective of rotor velocity, Pulse Width Modulation Method imposed relation : Voltage / Frequency =constant.

Many great companies provide a single microcontroller, which own control functions while generating PWM waveforms in which the modulation is sinusoidal. Motorola's μ68332 (332) is particularly well-suited to controlling induction motors. The 332 is built around the CPU 32, a 32-bit core whose instruction set is almost identical to the μ68020's.But here, the waveforms have the same period (and, of cause, the same frequency !), independently varying pulse widths, and well-controlled time relationships among phases. That is one deficiency. Another is the specific assembler language for Motorola's μ68332 (332) or μ68020's microcontroler, which is not comfortable for an customer familiar with I.B.M.-P.C assembler language or with C program. The second deficiency is the price of microcontroler, in which is not included a Proportional Integration Derivation (P.I.D) Regulator and so, the price must be added to that.
At last, the last deficiency is the reliability of such a fixture, reliability much smaller than a virtual P.I.D., included in a C++ program, which program is much steadier and offers a much more friendly and engaging graphic interface than any other assembler language. Those are performed by our program. Another known cip is ASIC Chip Incorporates 8-bit RISC Core And

Associated Firmware To Precisely Control Single-Phase Motors. By its name says, three cips are needed for to control an 3-phase induction motor and it should be difficult to synchronize them together. Concerning Digital Signal Processors (D.S.P), a very good one of them is TMS320C50, which may compute P.W.M. and the estimation of speed, but it also has its own assembler language, difficult to be learned. Our C++ program.exe might be written in a microcontroler or a D.C.P, single or included in a macro-program, by principle "hardware for software".

II. INVERTER FED INDUCTION MOTOR SCHEME

Figure 1 shows a bipolar transistor inverter block scheme. Below, more details will appear.

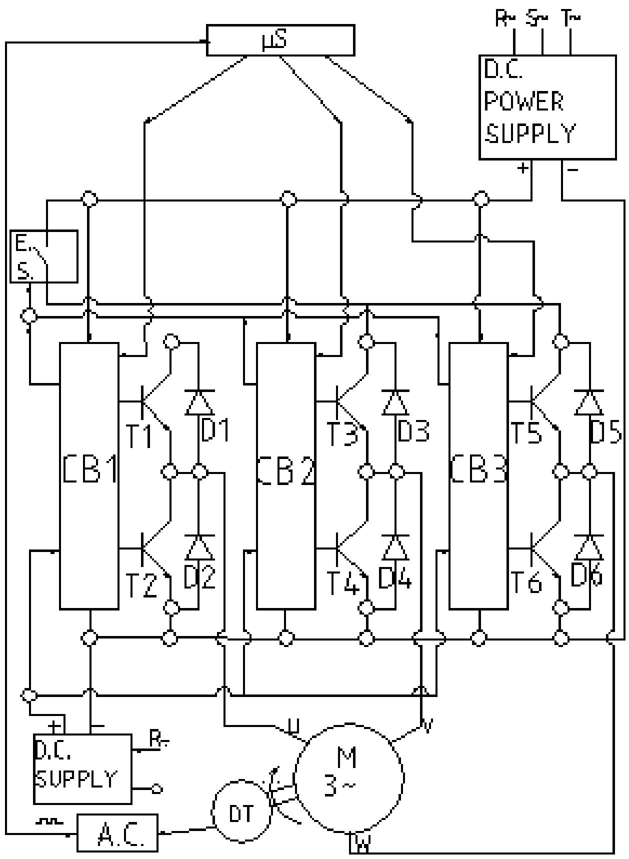


Fig.1.Bipolar transistors inverter block scheme

The three phases necessary for the motor are marked down with u, v, w. and the three line voltages between them, desirable as approaching of a sin as much it is possible, are generated by switching six power bipolar transistors T1…T6. Diodes D1…D6 retrieve the electromagnetic energy back to DC POWER SUPPLY, energy obtained from motor, during a decreasing speed motor process, commanded by the customer. DC POWER SUPPLY is supplied by three phases R. S. T standard 380 V / 50 Hz and, at its turn, delivers d.c. voltages for six inverter power transistors T1…T6., but also for Command Blocks CB!…CB3, designated to command those

transistors. E.S. designates an Emergency Scheme, which task is to cut out DC POWER SUPPLY, only from power transistors, when one or many of the Command Blocks deliver a wrong command for transistors. It means that two transistors on the same vertical line are simultaneously in conduction. The second D.C. SUPPLY, situated in the bottom-left side of Fig.1, is used for supplying logical cips, which interface the µP with LED-s of optoncouplers.. They electronicaly isolate the µP parallel and serial ports from the inverter, supplied by high d.c. voltages, which may ruin the first. DT is an incremental speed transducer, whose pulses are coded by an Acquisition Cart (A.C) and spent to µP.
Using BUX 82 as power transistors, with V_{ceo} = 800 V, Ic= 10 A and f_T = 1 Mz, by databook maximum ratings, a maximum 2 Kw electrical power induction motor may be supplied, because dissipated power on transistors must be taken in account.

If more power is needed, the scheme in figure 2 would be used, where high-speed and power insulated-gate bipolar transistors (IGBTs type SKM 300 GA 102 D) allow large variable-speed drives to operate at 20 kHz frequency and 1,750 Kw dissipated power, with V_{ces} = 1000 V, Ic= 300 A, which supplies a maximum 100 Kw electrical power induction motor. This is a substantial improvement over drives built from conventional bipolar devices. Figure 2 shows such a scheme.

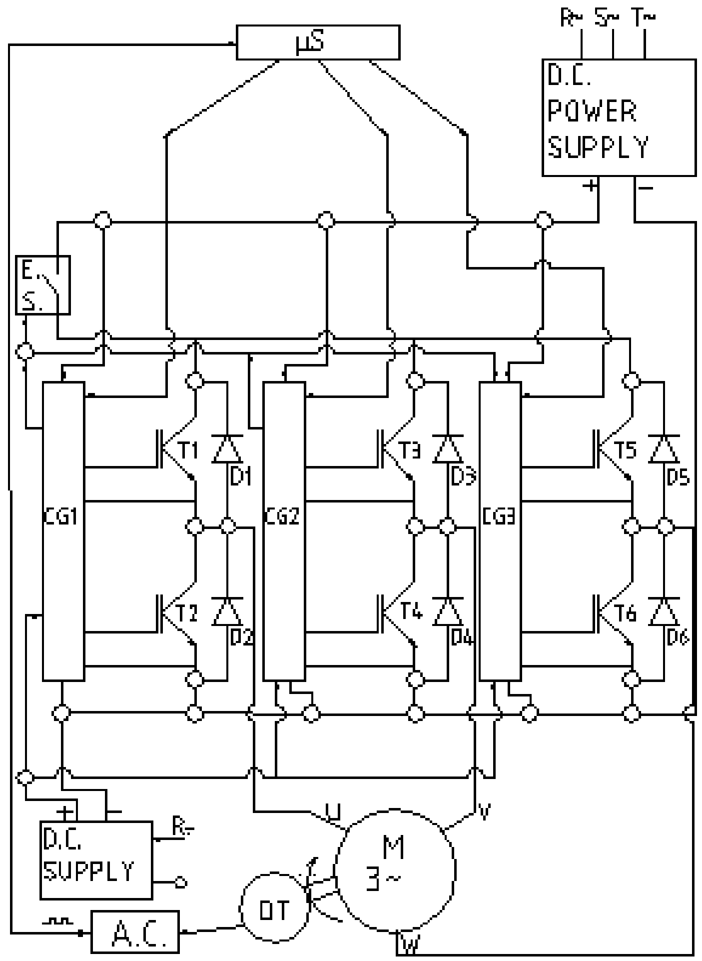


Fig. 2. I.G.B.T's inverter block scheme

The only difference between figure 1 and 2 is the driving blocks for high power transistors, the Command Gates blocks (C.B) of figure 2 containing a scheme for to charge the capacitor C_{gs} (Gate-Source), scheme with a very small internal resistor, the goal being a very short switch-on time for the I.G.B.T.'s. This scheme performs also a very short switch-off time, by discharging the same capacitor on a very small simulated resistor. A detail in figure 1 above, Command Blocks I (C.Bi, i=1,2,3), is shown below, in figure 3 :

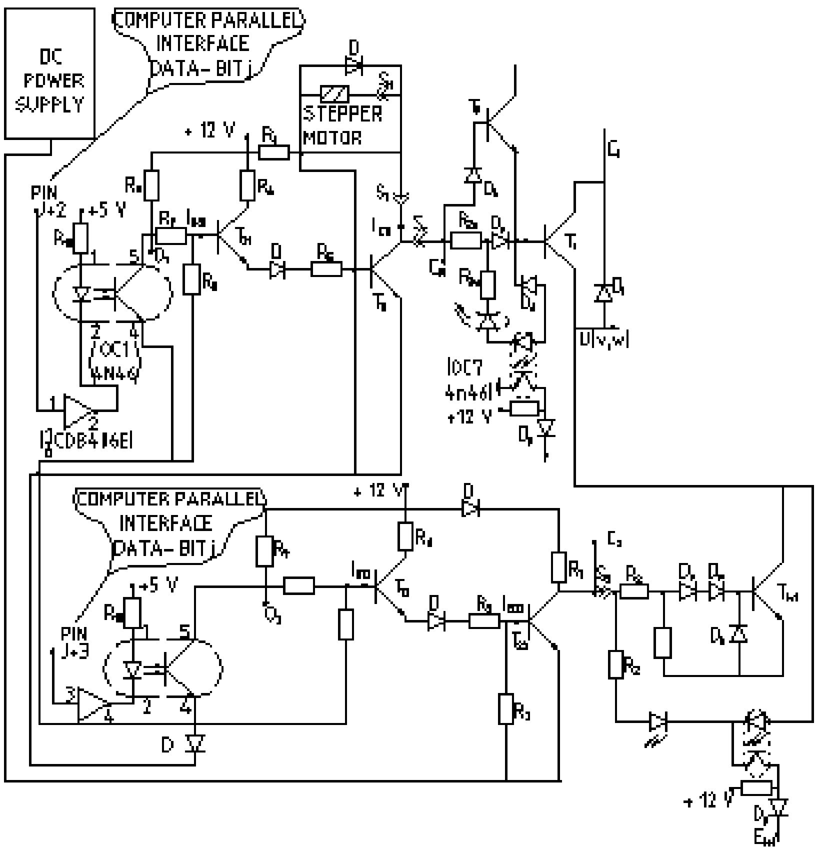


Figure 3. Details over command inverter scheme

T_{11} and T_{21}, on the vertical line are BU 208 switch type, the command drives for power ones, T_1 and T_2, which are the same as in figure 1 above and are BUX 82 power type. T_{1p}, in the top-left corner determines a deep saturation for T_1. The optocouplers, attached in the T_{11} and T_{21} bases, are used in Emergency Scheme (E.S., see figure 1 and 2), for to protect the inverter in the dangerous situation when both bases of pair T_1,T_2 (or T_3,T_4, respective T_5,T_6, in figure 1), are in up state. The Acquisition Card (A.C-see Figures 1 and 2) is, in fact, a mouse device, installed on the external side of the motor ax, mouse coupled in the PS/2 serial computer interface. Mouse driver functions are accessed through interrupt 33h. The inverter was built only to prove the good running of the C++ program and is not intended to be special or new, but the program is.

III. C++ PROGRAM FOR TO COMMAND AN INVERTER FED INDUCTION MACHINE

This program directly implements PWM by generating all three PWM signals and provides control functions, communications, and a user interface. It is composed by source program and header. In the first, interrupt-driven cause logic-level transitions on a given output pin, that produce PWM waveforms having a specific pulse width and period. On the occurrence of a predetermined timer-match condition, the pin state makes the appropriate transition, and the program calculates the next low-high and high-low transition times.

```
Void interrupt handler(__CPPARGS) //the timer interrupt routine (for int 8)
{
 if(!stopflag) //if the motor is started ,then wait for a transition on a pulse (high/low or L/H)
  {
  if (!move) //wait for the first transition
            {
   newimp=inport(0x379); //read status port, where only a bit position is tied to the transducer.
.
.
orig:   //this section calls the original interrupt routine with  a frequency of 18.2 Hz (the original frequency)
  asm{
          pushf
          call original
          mov al,0x20 //we need to write...
          // the EOI command to the port 20h of 8259 circuit
          out 0x20,al
          }
endint
}
```

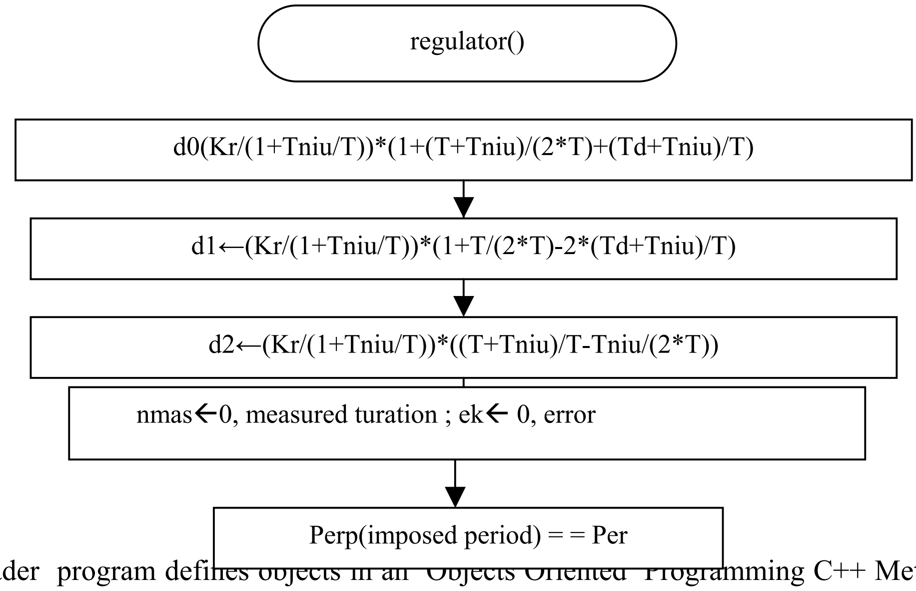
Because ac induction-motor control requires three PWM signals, this technique imposes a service time that severely limits the maximum PWM frequency a µP can produce. Such software latency is highly undesirable because it results in PWM frequencies within the audible range.

```
original=_dos_getvect(8);  //get the original vector for int 8
 _dos_setvect(8,handler); //set the new interrupt function for int 8
  asm{
      mov ax,intval
      out 0x40,al
      mov al,ah
      out 0x40,al
     }  //sets the timer interrupt occurrence at every 0.1 ms
```

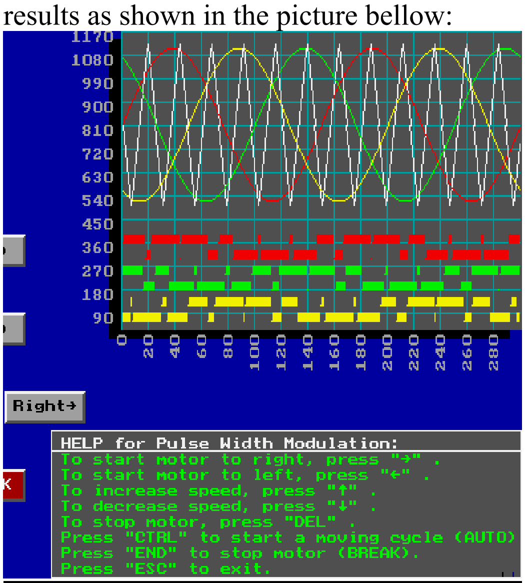
Calculations are effectuated by the Floating Point functions, as is shown bellow:

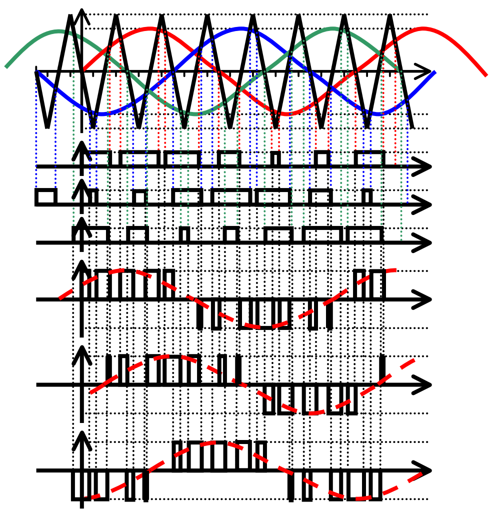
```
double f(int j) //the sine function
{

double Amplitude=Amp;
 double Period=Per;
 double Phase=fi;
 double two=2.;
 int t=j;
 double rez;
asm{
   fldpi
   fmul two
   fdiv Period
   fimul t
   fadd Phase
   fsin
   fmul Amplitude
   fstp rez
   }
return rez;
}
```

P.I.D. regulator gives parameters for those :



The Header program defines objects in an Objects Oriented Programming C++ Method, with results as shown in the picture bellow:





IV.CONCLUSION

The paper shows the results of two years work. The next step is the implementation in C++ of the rotor Field Oriented Method.

V. REFERENCES>

[1]Frank Hoffmann, Stefan Koch: *Steady state of Speed Sensorless Control of Induction Machines, Ruhr University's Symposium, 1998*

[2]Octavian Catrina, Iuliana Cojocaru: *Turbo C++,Editura Teora, 1994*.

[3]Alexandru Fransua, Razvan Magureanu: *Masini si Actionari Electrice, Edita Tehnica, 1986*

[4]Arpad Kelemen, Maria Imecs:Sisteme de Reglare cu Orientare dupa Camp ale Masinilor de Curent Alternativ, Editura Academiei, 1989