CONTROLLER AND ACTUATOR OF THREE INDEPENDENT DC MOTORS IN CLOSED LOOP

Gil Lopes , Fernando Ribeiro, Bruno Matos,
gil@dei.uminho.pt, fernando@dei.uminho.pt, brunoggmatos@gmail.com

Universidade do Minho, Departamento de Electrónica Industrial, Guimarães, Portugal

Abstract – This article describes a solution for a high power controller and actuator of three DC motors in closed loop. This controller can be applied to omnidirectional platform solutions using three motorised Swedish wheels as used by several RobCup MSL robots and Minho omnidirectional Wheelchair. Some existing controllers in the market are reviewed pointing out their characteristics and comparing them with the proposed solution. Operational characteristics and developed algorithms of the proposed system are fully disclosed.

Keywords – Encoders, PID controllers, DC Motors, Sensors.

1. INTRODUCTION

Mobile robots have locomotion systems that can vary in shape, size and type. Several concepts of locomotion can be found in the market today [1] and they are utilised according to their needs. The two major base systems are legs and wheels and each of these systems have several variations of their principle. This work is about wheel locomotion systems and in particular of omnidirectional type.

Fig. 1 shows a three omnidirectional wheel platform. These wheels (also known as Swedish wheels) are displaced 120º of each other as shown in the figure thus enabling their omnidirectional movement as a function of the rotational power supplied to each wheel. This kind of platform can move in any direction including the rotational movement on its own hence increasing their usefulness to reach certain locations otherwise impossible to other types of locomotion. Most RoboCup MSL teams and Minho omnidirectional wheelchair use this technique for robot locomotion (see Fig. 2).

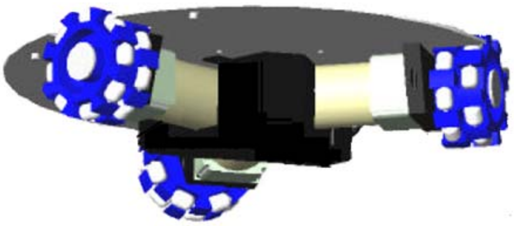


Fig. 1. Omnidirectional platform

When powered by batteries these platforms can operate wirelessly. It gives them all freedom of movement to take the most out of this type of locomotion. These three wheel platforms are commonly driven by DC motors attached to each wheel. The sum of the vector speed of each motor will define the direction and speed of the platform's movement.

Fig. 2. Minho team RoboCup MSL robot (left) and omnidirectional wheelchair (right)

To control the speed of three motors a controller card is needed. This work is about the development of a motor controller that can operate in these conditions and be able to power the platform DC motors with encoder feedback. A set of initial parameters were defined to allow the development of the controller to be in line with the existing hardware. The controller should communicate to receive and send data with a master controller like a personal computer (PC) or other type of microprocessor device. This communication should be via I2C bus where controller parameters such as motor speed, PID parameters amongst others are sent to the controller and feedback values such as temperatures (motor and board power electronics) and encoder counts are sent back to the master device.

Other definitions were set such as the operating voltage of 24 V and a maximum motor power of 500 W. The controller operating frequency should be out of the audible region, low electromagnetic noise and provided with protection circuits for excessive currents, voltages and temperatures. After the requirements were set an initial solution was then proposed. Fig. 3 shows a diagram of the relevant parts that comprise the initial proposal.

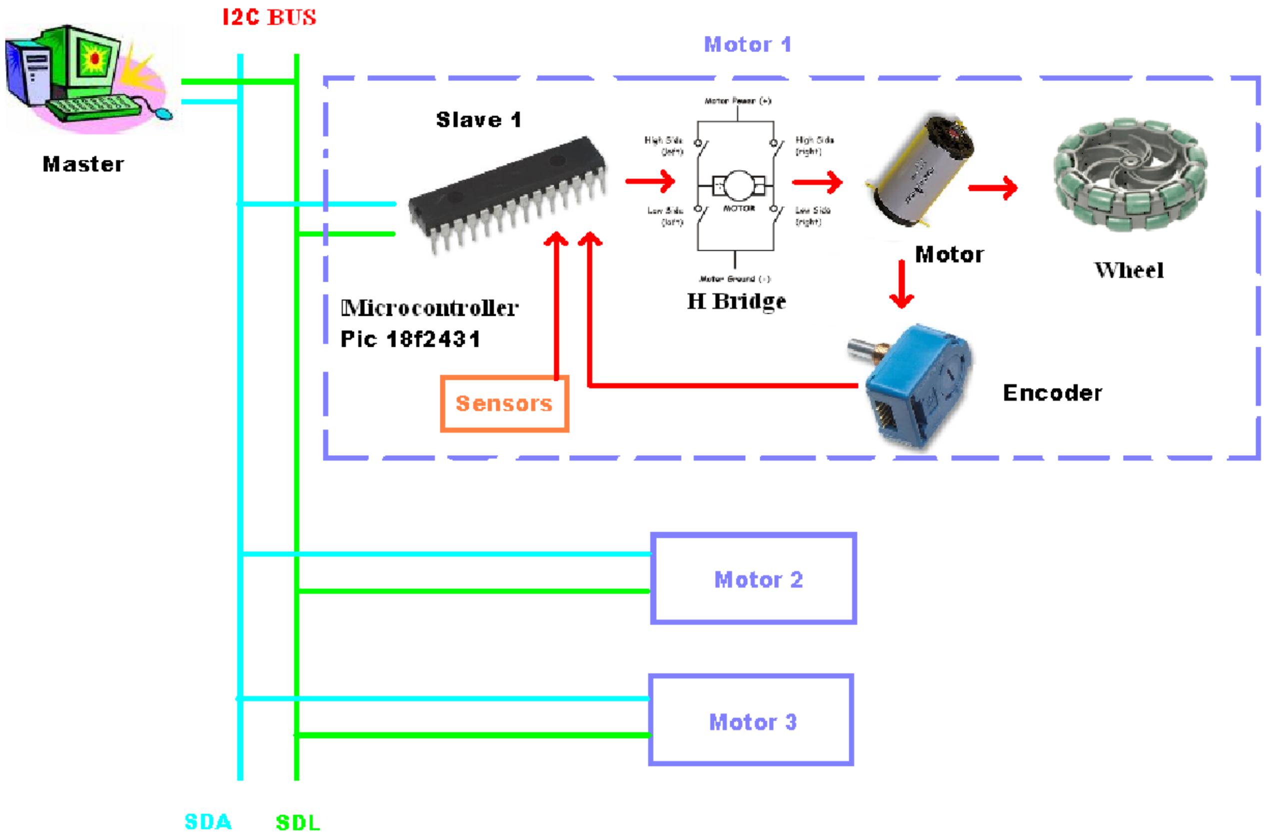


Fig. 3. Diagram of the proposed solution.

This proposal is based on three 8-bit PIC microcontrollers from Microchip, three H-bridges as motor drivers, sensors for temperature, current and voltages, with an operating PWM frequency at around 19500 Hz. MPLAB C18 was the software chosen to develop the PIC software using C and assembly languages. A possible approach is to build three separate controllers instead a single one to control all the three motors at once. This proposal was preferred rather than a single solution because it gives flexibility to reuse this board for single motor solutions. Moreover this subject is discussed and the decisions taken are explained in more detail.

2. MARKET SURVEY

Motion control systems are often complex and expensive. For most applications there is a variety of controllers each of them with their advantages and disadvantages. Some controllers will be here presented and their details discussed.

TMC200 is a motion controller system from Fraunhofer Institut fur Autonome Intelligente Systeme (AIS) [2]. The controller is shown in Fig. 4. It is able to control three DC motors in closed loop of 200W each. With a single 16-bit microcontroller, it uses PID control and feedback readings from an encoder to assure the right motor speed. This device communicates via CAN and RS232.

Fig. 4. TMC200 motion controller from Fraunhofer AIS

This controller is quite functional and compact but has no I2C communication and a maximum motor power of 200W. Its advantage though is the control of three DC motors from a single controller.

ADS 50/10 [3] is a motion controller from Maxon and shown in Fig. 5. This device can power and control a single DC motor of up to 500 W and a limited current of 10 A. Motor speed is adjusted by potentiometer. It operates with encoder feedback using proportional only PID control.

Due to its speed adjustment by analog variation it does not fulfil the requirements for a digital communication via I2C without hardware development for the adaption. This type of controller is more suitable to manual speed adjustment applications such as conveyor belts in industry, etc. Robotic football speed variations can be performed at a rate of up to 30 times per second.

Fig. 5. Maxon ADS 50/10 motion controller

Fig. 6 shows the MD03 from Devantech [4]. This motor controller can power and control a DC motor of up to 1000W. It uses an I2C bus for communication, has protection against short-circuits and over temperatures.

Fig. 6. Devantech MD03 motion controller

This powerful motor controller lacks on feedback control input. It has no connection for encoders or tachometers to supply input to the PWM controller in order to adjust the correct motor rotational speed. An external circuit would be necessary to close the loop on the control side.

Acroname S24-15A-30V is a 450 W motor controller. This controller shown in Fig. 7 does not have an I2C bus for communication and motor speed adjustment is attained by the user's external PWM. The quadrature encoder pins serves only as pass through pins thus no internal motion control is available [5].

Some more motor controllers were found in the market such as the MD22 [6], Simple-H [7], µM-H-Bridge [8], RoboteQ AX1500 [9] but they all demonstrate the same disadvantages from the previous shown controllers. None of them fulfils the total required parameters. Most of the presented controllers also do not have sensors for over temperatures, short-circuiting and current overload.

In order to have all the necessary requirements, external hardware would have to be developed increasing the total final cost of a working system. Instead, it was opted to develop a complete system from scratch.

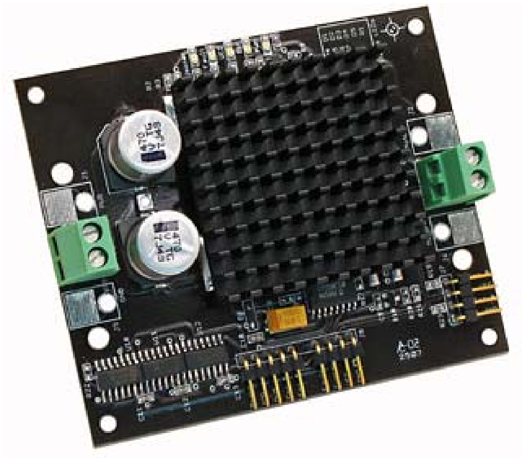


Fig. 7. Acroname S24-15A-30V motor controller

3. PROPOSED SYSTEM

To elaborate the proposed system each part was studied in detail as it is shown in this chapter. The following step-by-step discussion describes the different hardware and software parts in detail as well as the effects caused on the system. The integral parts of the system are:

- Microcontroller PIC18f2431 from Microchip
- Quadrature Encoder inputs
- PWM
- PID control
- Bus I2C
- Sensors
- H Bridge

3.1 - Microcontroller

The chosen microcontroller is the brain of the whole system. It interfaces with a Master controller to receive commands, it interfaces with the H-Bridge generating the PWM for the motor, it reads and counts the pulses from the quadrature encoder and it implements a PID control to sustain the intended motor rotary speed.

Fig. 8 show a picture of the microcontroller used. Some of its relevant characteristics are [10]:

- Four 14-bit PWM generators
- Quadrature encoder inputs
- 10-bit ADC of 200 ksps
- 10 MIPS
- 2176 kB total memory

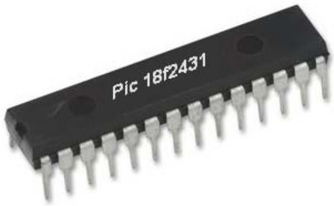


Fig. 8. Microchip PIC18f2431 microcontroller

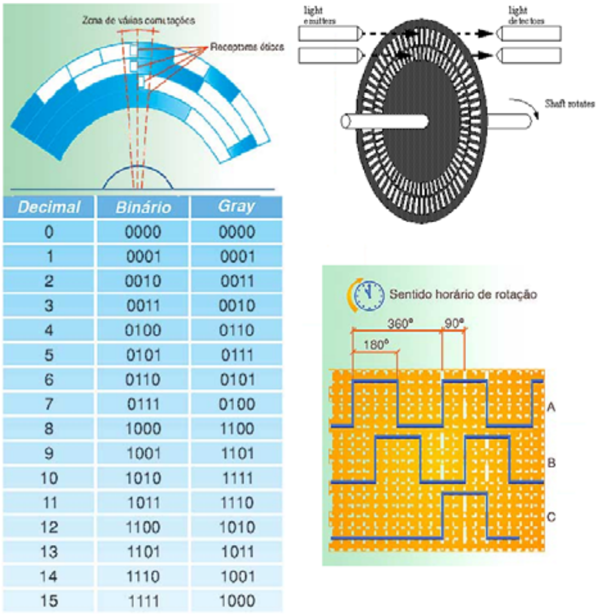
As mentioned earlier the solution shown in Fig. 3 uses three microcontrollers to control three motors, one per each motor. Some thoughts were taken about using a single controller to perform for three motors but it was found that the system could hang up due to processing overload. At the same time this solution is more flexible to be used standalone with a single motor for different applications.

3.2 - Encoder

Fig. 9 shows a common quadrature rotary encoder generally found attached to many DC motors. This device converts the angular position of a motor shaft into a digital code.

Fig. 9. Avago Technologies HEDS5700 encoder [11]

There are two types of encoders: incremental and absolute. On incremental encoders position is given by pulses from a start pulse zero with no relation to the shaft angle. On the other hand absolute encoders supply a unique code that is relative to each position of its course. Fig. 10 shows the differences between the two types of encoders.



Decimal	Binário	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Fig. 10. Absolute encoder (left) and Incremental encoder (right) [12]

Incremental encoder was then defined to be used for this work. It is known to be a quadrature incremental encoder thus creating two pulses separated 90º from each other as shown in Fig. 10 (right). It is usually called channel A and channel B. This allows the system to detect which way is the shaft rotation (clockwise or counterclockwise). A reading of only one channel provides the rotational speed, whereas the reading of the two channels also provides the direction of movement.

Another signal called Z or zero is sometimes available in some encoders. It gives an absolute zero of the encoder. This signal is a square pulse in which the phase and width of the channel are the same [12].

To determine de direction of rotation the microcontroller should read channel A and channel B and implement a D flip-flop as shown in Fig. 11. If the pulse on channel A comes first than channel B then the movement is clockwise, otherwise it is counterclockwise.

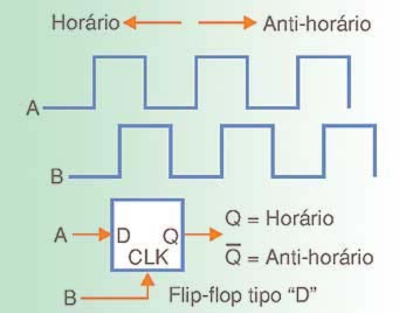


Fig. 11. Direction of rotation on encoder signals [12]

To determine the speed of rotation two methods can be used. The first is to count *n* pulses in a fixed time interval *t*. This would be the minimum time to detect one pulse at one rotation per minute (rpm) with an encoder of *k* pulses per revolution (PPR) (1).

$$t = \frac{60}{k}. \tag{1}$$

Knowing that *n* pulses have occurred in *t* time then the speed can be determined as (2).

$$rpm = \frac{60 \times n}{t \times k}. \tag{2}$$

The ranges of updating speed are inversely proportional to the encoder's PPR value and the cost of the encoder is proportional to the PPR value. Therefore this solution was not chosen.

The option chosen is to measure the time that a pulse generated by a channel is in the logical level 1 or 0. This implementation is complex due to time measurement accuracy in relation to low resolution of the timer. This is also valid when measuring the difference in time for close speeds. As an example, for the microcontroller oscillator clock at 40MHz the timer increases every 100 ns. For a 500 PPR encoder with the motor at 6999 rpm the time necessary to measure one logical level is 8.5727 µs. For 7000 rpm is 8.5714 µs. The difference is 1.22 ns that is much lower than the 100 ns resolution of the timer.

The solution found was to increase the number of pulses for high speeds and to use a prescaler for low speeds also reducing the number of read pulses. Table II shows the number of pulses read a function of speed. Speed values were calculated with a margin of ≈ 50%.

Table 1

Postscaler Rpm	64 Pulses Timer/1		16 Pulses Timer/1		4 Pulses Timer/1		1 Pulse Timer/8	
<=127						🟩	🟩	🟩
128 -510				🟩	🟩	🟩	🟩	
511-2000		🟩	🟩	🟩	🟩			
>2000	🟩	🟩	🟩					

Equation (3) calculates the rotation speed as a function of:

a) postscaler - number of pulses read

b) VELRH and VELRL - registers of the time read by timer.

$$rpm = \frac{30 \times potscaler \times processor\_clock}{(VELRH \times 256 + VELRL) \times PPR \times prescaler} \tag{3}$$

3.3 – Pulse Width Modulation - PWM

PWM is a method to vary the amount of power supplied to a motor in order to change its rotational speed. By varying the pulse time ON (duty cycle) in relation to time OFF the average supplied power changes as shown in Fig. 12. This motor speed variation method is preferred to voltage variation mainly due to very low torque at low speeds. Several advantages are found when using PWM: a) it can be easily generated in a digital form by a microcontroller; b) the losses are minimal; c) the actuator element (usually IGBT, MOSFET or bipolar transistor) operate in the state of cut/saturation and the loss solely on the commutation [13].

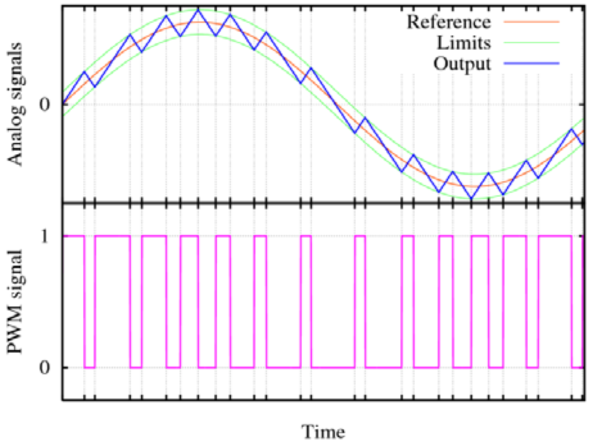


Fig. 12. PWM signal [14]

According to the microcontroller datasheet PWM resolution is the number of bits that defines the duty cycle and working frequency is the base PWM frequency that can be generated by the microcontroller. Since one of the initial requirements of this work is PWM operation outside the audible frequency for a human being and based on the microcontroller datasheet about resolution and frequency it was decided that the working frequency would be 19500 kHz with 11-bit resolution.

3.4 – H-Bridge

H-Bridge is an electronic circuit that allows a simple reversing mechanism of a DC motor rotation direction. It is made of four elements (usually transistors) that are saturated two at a time as shown in Fig. 13. When Q1 and Q4 are closed a circuit is made making the motor to rotate clockwise. On Q2 and Q3 closed the motor spins counterclockwise. Along with PWM this is a full digital solution to operate a DC motor in terms of speed variation and direction settings. The H-bridge though is the only power circuit on a DC motor drive solution. It is also what defines the maximum power allowed on the system.

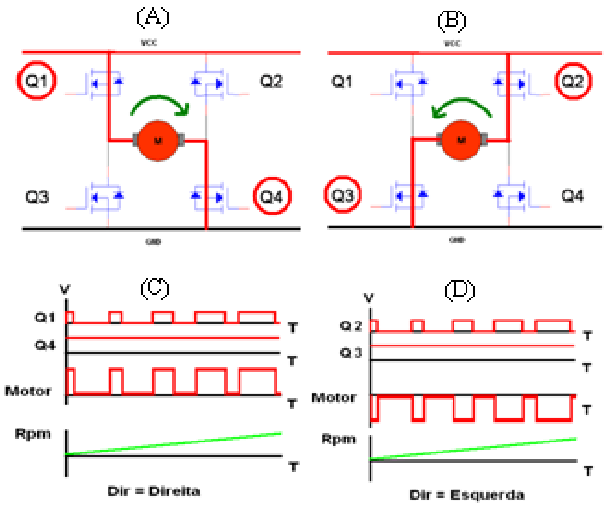


Fig. 13. H-bridge operation (top) and respective signals (bottom)

3.5 - Controller

The controller is responsible for executing what is asked for: set and keep a steady motor speed. It also corrects disturbances that may exist in the system. When a new speed is set the controller adjusts the PWM duty cycle according to the actual speed using the error value. This is based on the speed difference between actual and new speed. The error reduces according to a control law that defines the process on how to change from a speed to the other.

The control law is an algorithm and different algorithms are known such as Phase Locked Loop (PLL), ON-OFF and Proportional-Integral-Derivative (PID). The latter is the most common method [15] and is shown in Fig. 14. PID control can be used in separate ways: a) only P; b) PI; c) PID.

Table 2 [16] provides a comparison of advantages and disadvantages of each control method. As it can be seen the PID controller provides better performance / price. It was the chosen algorithm to be implemented in the microcontroller.

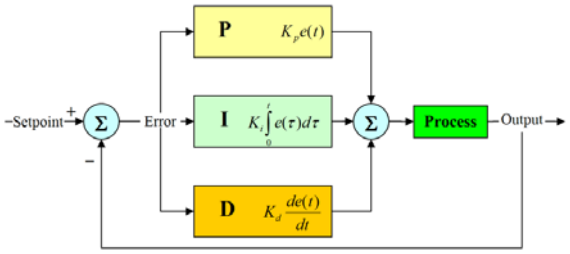


Fig. 14. Block diagram of a PID controller [15]

Table 2

		Control			
		P	PI	PID	PLL
Precision	High				🟩
	Med		🟩	🟩	
	Low	🟩			
Speed	High	🟩		🟩	
	Low		🟩		
Good in low rotation		🟩		🟩	
Cost	High				🟩
	Med			🟩	
	Low	🟩	🟩		

3.6 - I2C Bus

I2C protocol was created by Philips and allows data communication in both directions with a transfer rate of up to 3.4 Mbit/s. To add a device the user connects it to the bus and gives it an address. It is a Master / Slave protocol in which the Master always has priority of communication to the Slave. Fig. 15 shows how devices are connected to the bus.

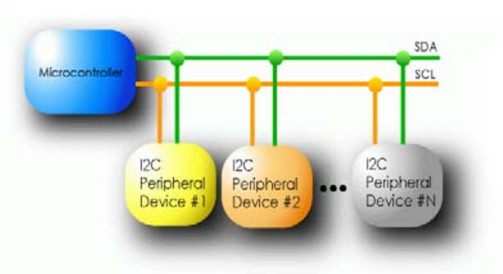


Fig. 15. I2C interface between Master and Slaves [17]

In this work the motor controller is set as a slave and the bus is used for receiving speed commands and PID settings from a master device (PC). The slave can also send back as requested by the master local information such as actual speed, voltages, currents, temperatures and PID controller settings.

3.7 - Sensors

Sensors are used to indicate system operation status. :

a) Current - Measures the current used by the motor, prevents short-circuits and detects motor absence
b) Voltage - Measures voltage supply for acceptance levels
c) Temperature - Measures motor and H-bridge temperatures to avoid over temperatures that may damage them

Temperature sensors use I2C protocol to communicate with the microcontroller whereas current and voltage sensors are embedded in the microcontroller.

4. TESTS AND RESULTS

Electronic circuitry was developed and two Printed Circuit Boards (PCB) were created. One contains the power elements of the motor controller and the other the communication and control parts. In this way it is possible to replace one or another for repairing or for improvement without replacing the whole circuitry.

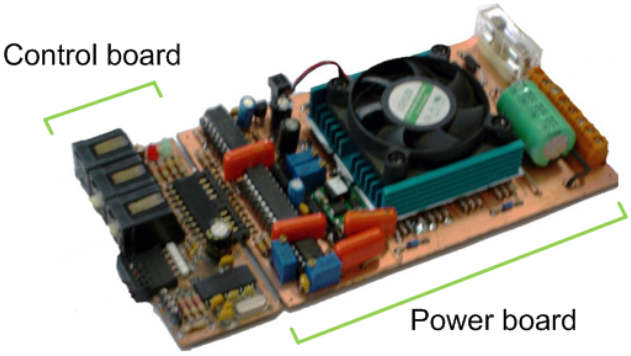


Fig. 16 – Developed PCBs: control board and power board

Experiments were conducted in order to test the boards to their defined limits. In the absence of a 500 W power DC motor a combination of two motors and two generators were used instead. A 200 W motor with an incremental encoder was the main motor and feedback to the system. A second 350 W motor was attached in parallel with the first motor and a load was belt connected to two 120 W generators. They supplied an adjustable load to the 350 W power motor to create a total output load to the controller board not less than 500 W. Measurements were made and confirmed the output power values expected. Fig. 17 shows a pictorial diagram of the test bench.

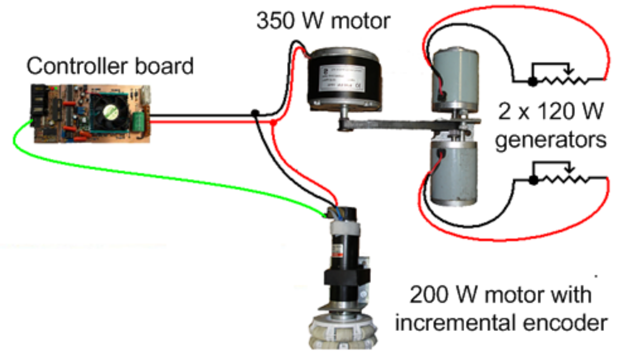


Fig. 17 – 500 W motor test bench

First tests were made varying the motor speed in steps of 1000 RPMs. Real motor speed was measured and a comparison with the intended speed is shown in Fig. 18. As it can be seen real speed follows closely to intended speed up to maximum rotation speed of the motor.

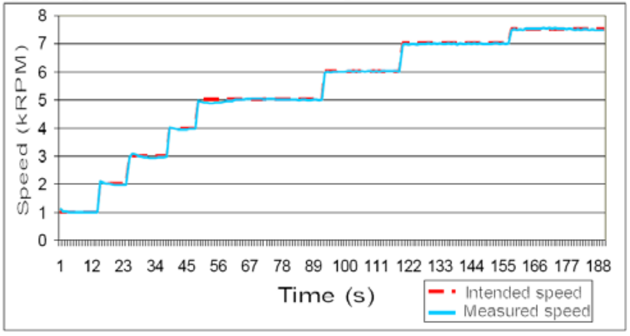


Fig. 18 – Motor speed response with load

A small overshoot can be seen on lower speeds due to a lower motor load at these low speeds. As the speed increases the whole motor inertia tends to be relieved and the overshoot disappears.

The graph of Fig. 19 shows voltage and current responses as the motor speed is increased. Voltage drops slightly when current increases above 5 A and a second small drop is seen when current goes above 15 A. The test bench system was powered by batteries and this variation was attributed part to a drop on the battery voltages and to the MOSFET junction at these currents. Since it was not a major drop and did not interfere with the system's performance further investigation was not considered necessary.

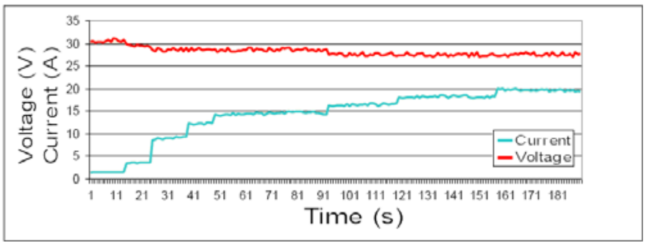


Fig. 19 – Voltage and current response to speed variation

The graph of Fig. 20 shows the measured output power from the controller. As it can be seen at maximum speed the system outputs more than 500 W of power. There is a safety margin of 150 W above the 500 W for the H-bridge transistors that should not be overtaken.

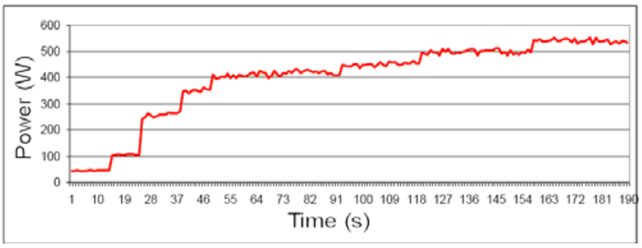


Fig. 20 – Power evolution with speed variation

5. CONCLUSION

Powering and controlling three independent DC motors in closed loop was the main objective of this work. Due to motor power definition and demands, circuit simplicity and flexibility and processing power it was found that a separate circuit per motor would achieve better results. After studying the various constituent parts of a possible system one was developed and tested successfully with the incorporation of safety circuitry.

Communicating via I2C to send commands and receive feedback information this system is easily integrated on a triple board solution to control omnidirectional mobile robots with three Swedish wheels. That was also tested successfully. It supports DC motors with incremental quadrature encoders of up to 500 W with voltages ranging 12 to 56 V. Maximum tested motor speeds was 7200 RPM. From the market analysis this system has resulted in a low cost, powerful, fast and accurate solution that can be applied broadly in many different motor applications. Future work will be to develop a master control system to control these three developed boards in order to alleviate the PC processing on computing the motion equations of the omnidirectional system.

REFERENCES

1. Böttcher, S., Principles of robot locomotion - Seminar. 2004, SIUC - Southern Illianois University Carbondale: Illianois - USA.
2. Surmann, H. TMC200 motion controller board. [cited 2008; Available from: http://www.volksbot.de/.
3. Motors, M. Maxon 4-Q-DC Servoamplifier ADS 50/10. 2009; Available from: http://test.maxonmotor.com/docsx/Download/Product/Pdf/ads50-10_en.pdf.
4. Devantech. MD03 - 24Volt 20Amp H Bridge Motor Drive 2008; Available from: http://www.robot-electronics.co.uk/htm/md03tech.htm
5. Acroname. High Capacity H-Bridge - S24-15A-30V-HBRIDGE. 2008 [cited 2008; Available from: http://www.acroname.com/robotics/parts/S24-15A-30V-HBRIDGE.pdf.
6. Devantech. Devantech MD22 motor driver. 2008 [cited 2008; Available from: http://www.acroname.com/robotics/parts/R220-MD22.html.
7. RobotPower. Robot Power Simple-H. 2008 [cited 2008; Available from: http://www.robotpower.com/products/simple-h_info.html.
8. ModularCircuits. µModule H-Bridge module. 2008 [cited 2008; Available from: http://www.modularcircuits.com/h-bridge.htm.
9. RoboteQ. RoboteQ AX1500 - 2 x 30 SmartAmps 12V-40V Robot Controller. 2008 [cited 2008; Available from: http://www.robotshop.ca/home/suppliers/RoboteQ-en/roboteq-ax1500-robot-controller.html
10. Microchip. PIC18F2431. 2008 [cited 2008; Available from: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010271.
11. Avago-Technologies. HEDS5700 quadrature encoder. 2008 [cited 2008; Available from: http://www.avagotech.com/pages/support/?id=954.
12. Mecatrônica-Actual. Como funcionam os Encoders. 2008 [cited 2008; Available from: http://www.mecatronicaatual.com.br/secoes/leitura/617.
13. Catsoulis, J. Motor Control and PWM. 2009 [cited 2009; Available from: http://www.embedded.com.au/pages/Motor_Interface.html.
14. Wikipedia. Pulse-width modulation. 2009 [cited 2009; Available from: http://en.wikipedia.org/wiki/Pulse-width_modulation.
15. Åström, K.J., PID Control, in Control System Design. 2002. p. 216-251.
16. Capeli, A., Controle Para Automação Industrial, in Saber Eletrônica. Feb. 2004, no. 373.
17. Soffel, V. Synchronous Microcontroller Communication Interfaces: SPI, Microwire and I2C Protocol Formats. 2008 [cited 2008; Available from: http://www.ucpros.com/work%20samples/Microcontroller%20Communication%20Interfaces%202.htm.