Control of Position/Velocity in a Mobile Robot Using DC Brushless Motors

Leonardo Romero and Antonio Concha
Universidad Michoacana de San Nicolas de Hidalgo
Morelia Michoacan 58000, Mexico
lromero@umich.mx, aconcha@faraday.fie.umich.mx

Abstract
This paper describes the construction of a differential mobile robot using a brushless DC motor coupled to each wheel. Considering that commercial controllers of brushless motors are expensive and they control only velocity, not position; we design and built 3-Phase Bridges, with N-Mosfets, within a electronic circuitry to drive brushless motors. A PWM control scheme and the outputs of the optical encoder and Hall sensor of the motor are used to implement a closed–loop velocity and position control. The real time control of the two traction wheels runs on a 68HCS12 microcontroller and the mobile robot accepts commands using an standard RS232 serial connection. The hardware design and software of this robot is available online.

1. Introduction
Nowadays mobile robots are in research laboratories as well as in industry, hospitals, museums, and an important issue is to have a powerful and reliable mobile robot.

Unfortunately most Mobile Robotic research in Mexico is done using imported mobile robots. However, these robots have several disadvantages: 1) they are expensive, 2) maintenance is slow and expensive, and 3) normally they are closed systems, without flexibility to add new capabilities.

To overcome this disadvantages, we develop an open, flexible, reliable, powerful and expandable differential mobile robot, designed and built locally. It is based on the Linux operating system and commercial hardware for sensors and actuators. All the software required by the robot was already part of Linux, or it was developed using tools under the GNU General Public License. In other words, there is no proprietary code.

In a previous version of the robot [5] we use Pittman brush DC gearmotors, which include quadrature encoders and a gear–box. Unfortunately the gears (of the gear–box) wear out very soon and they introduce significant position errors. Also brush DC motors generates electrical noise and brushes also wear out in a short time. A good reference to hardware design for brush motors in mobile robotics is also found in [1].

Our new mobile robot inherits its reliability from brushless DC motors and gearboxes made with industrial gears. The robot follows a differential scheme [2] with two wheels on a common axis, each wheel driven independently by a brushless motor, and two caster wheels to ensure balance. Figure 1 shows a view of the motor with its gearbox. Brushless motors are more reliable and less noisy than brush motors and have a longer life–time. However, because brushless DC motors has no brushes, they need an external electronic driver. Commercial driver cards or chips are very expensive, and they usually cost more than the brushless motors. In our case, we use the 4441S010 brushless motor from Pittman, which costs 208 USD. As a reference, the commercial velocity controller for a single motor, using Hall sensors or quadrature encoder, costs about 358 USD ¹.

¹http://www.clickautomation.com
of the brushless motor controller. The rest of this paper is organized as follows. Section 2 presents how brushless DC motors work. Section 3 presents the hardware design of the 3-Phase Bridge required, including also the interface with the microcontroller. Experimental results using the new mobile robot is shown in Section 4. Finally, some conclusions are given in Section 5.

2. Brushless DC Motors

Figure 2 helps to understand the operation of a brushless DC motor. The motor has 3 power inputs, labeled A, B and C. The 3-Phase Bridge has 6 switches, AL, BL, and CL in the low side and AH, BH and CH in the high side. The brushless motor also have a magnetic Hall sensor with 3 digital outputs, that encode the position of the shaft of the motor. The driver of the motor should see the output of the Hall sensor and close only one switch of the low side of the 3-Phase bridge and only one switch of the high side. Once the motor rotates, the Hall sensor reports new outputs and the driver should open the switches and then close another pair of switches.

![3-Phase Bridge and Over Voltage Snubber](image)

**Figure 2. Operation of a brushless motor.**

Whether using solid state switches or relays, problems arise when switching inductive load such as motors. We know that the voltage, \( v \), induced across an inductor is proportional to the rate of change of the current, \( i \), through it, and the inductance, \( L \); \( v = L \frac{di}{dt} \). If the current through an inductor has reached a steady state, the voltage across it is 0V. To prevent a high voltage spike (that can destroy the switch) when the current is upset by the opening of the switch, usually fly back diodes in the reverse direction across the switches are used to create a return path for the current. For instance, if \( AH \) and \( AL \) are closed, current flows through the motor from \( A \) to \( B \). When both switches are opened, the current flows through the diodes of \( AL, BH \) and the battery. In this way the power will fly back to the battery [2] and to the over voltage snubber.

An advantage of the brushless DC motor over the brush DC motor is its ability to hold its position. It is enough to keep closed the same pair of switches. In other words, brushless DC motors are like step motors (12 steps per revolution in our case).

To control the velocity (or torque) of the motor a very common approach is to use the Pulse Width Modulation (PWM) scheme [2]. In this scheme, the pair of switches are opened and closed in a periodic sequence (v.gr. with a frequency of 22KHz in our case). If switches are open almost all time, the velocity (or torque) is slow; but when switches remain closed more time, the velocity (or torque) increases. From now on, a high PWM means a PWM signal that cause a high velocity (or torque).

Now we are ready to describe our brushless motor controller.

3. Hardware

The mobile robot have two motors, two 3-phase bridges and one microcontroller. Now we discuss the implementation of the 3-phase bridge and its interface with the microcontroller. We also discuss the over voltage snubber, a very important part of the driver to reduce the temperature of the Power Mosfets and provide a safe operation when the motor rotates and also when the motor holds its position.

3.1. 3-Phase Bridge

Because brushless motor have a low resistance (3Ω in our case) and we use 24V (two lead–acid batteries 12V, 7Α-H), we have currents of several amperes and hence we need robust switches. Due its low resistance and low voltage drop, power Mosfets were chosen in the design of the 3-phase bridge. Also they have the additional advantage of have its own fly back diode.

We tested three versions. The first two versions, shown in Figure 3 and 4 have P-Mosfets in the high side and N-Mosfets in the low side. Both Figures only show the circuit for the \( A \) phase, because the other two phases are identical. Connections \( AH \) and \( AL \) receive the PWM scheme, are TTL compatible and they define the state of switches. The main drawback of version 1 is its long time to turn on (or close) or to turn off (or open) the Mosfet, since there is a high resistance in the circuit that feeds the Mosfet gate. Although it works fine when the motor is moving, Mosfets are too hot when the motor does not move and the PWM is
high. Version 2 uses Totem Pole outputs and hence have a faster switching operation. Now only the P-Mosfets get too hot when the motor does not move and the PWM is high. The problem is a drop voltage in the gate of the P-Mosfet due to the drop voltage in the power source when the Mosfet is turned on. In this circuit there is a different 12V reference voltage for the N-Mosfet (actually another 12V battery that also feeds the microcontroller).

The final version of the 3-Phase bridge, including the over voltage snubber, is shown in Figure 5. It uses six N-Mosfets, the IR2130 chip (a 3-Phase Bridge Driver) and the 74367 chip (3-State Buffer). A 15V regulated power source, provides the reference gate voltage for the IR2130. To implement the PWM scheme in all six Mosfets a single PWM signal, from the microcontroller, is connected to the enable pin (EN) of the 3-State Buffer. When the PWM signal is high, all Mosfets are turned off (using resistors connected to 5V). But when the PWM signal is low, the state of AL, AH, \cdots CH signals (from the microcontroller) control the state of Mosfets.

The final version also has the advantage of using only N-Mosfets which are cheaper and more accessible than similar P-Mosfets. Besides that, it has a 15V input to provide the required voltage of the N-Mosfets. In the implementation the 15V input is independent of the power source of motors, and hence it has no the drawback of version 2.

### Figure 3. Version 1.

![Figure 3. Version 1.](image)

### Figure 4. Version 2.

![Figure 4. Version 2.](image)

#### 3.2. Snubber Circuit

An over voltage snubber circuit (located at the right side of Figure 2) is incorporated in the power circuit to reduce the stress in the Power Mosfets during the switching and to assure a regime of safe work [4, 6]. Because there is a small bit of inductance in the wiring between the batteries and the driver, when the current suddenly stop, that inductance presents a higher voltage until somehow the current does manage to flow (through the Power Mosfets in our case when its maximum drain to source voltage is exceeded) [3]. Now with an RCD snubber, the capacitor absorbs that pulse of current through the diode. Once the current has decreased to zero the over voltage on the capacitor decreases to source voltage (24V) through the resistor.
3.3. Microcontroller

We use the Adapt9S12DP256 card, a compact, modular implementation of the Freescale 9S12DP256C microcontroller chip. It can run up to 24Mhz and two 50-pin connectors bring out all I/O pins of the microcontroller. It includes all necessary support circuitry for the microcontroller, as well as a 5-Volt regulator and two RS232 transceiver on-board.

This microcontroller can drive up to 8 PWM signals and 8 input captures, as well as 4 pulse accumulators. We use two PWM signals, one per motor. We also use two pulse accumulators, one per motor, to count pulses from the optical encoder of the motor. The velocity of the motor is computed from the number of pulses per millisecond.

The implementation of the control of both wheels, of the mobile robot, follows the control scheme presented in [2]. We only add an acceleration scheme like the one shown in Figure 6, to achieve a smooth movement. At the beginning, the robot moves slowly and the velocity is growing. Near the end of the movement, the velocity gradually decreases. The number of steps (and hence the position of the wheel) is easily computed from the signals of the Hall sensor.

To hold the position of the robot a simple proportional control, using the count of pulses per $\frac{1}{10} ms$ from the optical encoders, computes the right PWM signal sent to motors.

4. Experimental results

The program for the microcontroller was developed in the C language using a public domain compiler for the HCS12 microcontroller, available in Linux. More information about the new robot and older versions are available online (http://faraday.fie.umich.mx/~lromero).

We use two different power supplies. The first one, with two 12V batteries, to feed motors. The other one, with two 12V batteries provides the input to the 15V regulator of the 3-Phase bridge and also feeds a 5V regulator (which powers the microcontroller, Hall sensors and optical encoders).

The mobile robot moves smoothly as expected but the interesting case is when the robot require a force to hold its position. Tables 1 and 2 shows the temperature (in centigrade degrees) without and with the snubber, respectively, of the hottest Mosfet (without any heat dissipator) and the motor, for different PWM values. Higher PWM values means higher torque of motors and higher temperatures.

The comparison of both tables shows that the snubber avoids high temperatures in the Mosfets and it becomes an essential part of the driver. This behavior is explained when we see Figures 7 and 8, which show the input voltage of the driver without and with the snubber, respectively (closer...
views are shown in Figure 9 and 10). We can see how the transient part is smaller when the driver includes the snubber.

Figure 11 shows the case of a snubber using only the capacitor, without the diode and the resistor, as suggested in many literature about 3–Phase bridges. In this case, the capacitor is not a solution because there is a resonance, transient are longer than before and Mosfets get too hot in a few seconds.

To prevent high temperatures of motors, also PWM values are limited to 180.

Finally, because we use a PWM frequency of 22Khz there is no audible noise generated from motors.

A view of the 3-Phase bridge board to drive two motors is shown in Figure 12 (the snubber is not shown) and a top view of the mobile robot is shown in Figure 13.

5. Conclusions

We have presented a design of a mobile robot using brushless DC motors. They add reliability, and the ability to (easily) hold the position of the robot as well as to count the number of steps of the motor. They merge the high torque and velocity of brush motors and the ability to move in steps, like step motors.

Unlike commercial drives which only have velocity control, our design is well suited to velocity and position control. An additional advantage of our controller is that it is cheaper than the commercial one (800 USD vs 180 USD).

References

Table 1. Temperature of components when the motor does not move. Case without snubber

<table>
<thead>
<tr>
<th>PWM</th>
<th>Mosfets</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>31°</td>
<td>27°</td>
</tr>
<tr>
<td>150</td>
<td>39°</td>
<td>30°</td>
</tr>
<tr>
<td>180</td>
<td>90°</td>
<td>46°</td>
</tr>
<tr>
<td>210</td>
<td>&gt;120°</td>
<td>67°</td>
</tr>
</tbody>
</table>

Table 2. Temperature of components when the motor does not move. Case with snubber

<table>
<thead>
<tr>
<th>PWM</th>
<th>Mosfets</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>30°</td>
<td>27°</td>
</tr>
<tr>
<td>150</td>
<td>38°</td>
<td>30°</td>
</tr>
<tr>
<td>180</td>
<td>51°</td>
<td>46°</td>
</tr>
<tr>
<td>210</td>
<td>52°</td>
<td>67°</td>
</tr>
</tbody>
</table>


