Mechatronics Design of a Mecanum Wheeled Mobile Robot

Peter Xu

1. Introduction

The Mecanum wheel was designed in Sweden in 1975. Using four of these wheels provides omni-directional movement for a vehicle without needing a conventional steering system (Muir & Neumann, 1990; Dickerson & Lapin, 1991; Braunl, 1999; Navy, USA, 2002 and Lunze & Schmid, 2002). The wheel itself consists of a hub carrying a number of free moving rollers angled at 45° about the hub's circumference. The rollers are shaped such that the overall side profile of the wheel is circular. However, wheel slip is a common problem with the Mecanum wheel, particularly when the robot moves sidewise, as it has only one roller with a single point of ground contact at any one time. This severe slippage prevents the most popular dead-reckoning method, using rotary shaft encoders (Everett, 1995 and Borenstein et al, 1996), from being performed well on the Mecanum robot. To cope with the problem, visual dead-reckoning was used as a slip-resilient sensor (Giachetti et al, 1998; Nagatani et al, 2000 and Kraut, 2002). This technique, also used in optical mice, makes use of an on-board video-camera continuously capturing frames of the ground beneath and image processing hardware on the robot determining the speed and direction in which the current frame has moved relative to the previous frame thus allowing the speed and direction of that point of reference to be calculated. However, visual dead-reckoning using a single camera or optical mouse can not provide all three-degree-of-freedom positional information for robot navigation and motion control. Fixed line following is the simplest and most reliable solution, yet is also the most limiting. A physical line is marked on the ground along the path which the robot is to follow (Everett, 1995 and Borenstein et al, 1996). For a robot that is set up in a fixed location for a set task this system is effective but for a research robot with omni-directional capability this approach is seen to be a primitive, though still viable, option.

This chapter presents a research project recently completed at Massey University, New Zealand. The research started upon an existing omni-directional platform built of a box-like aluminium chassis, four electric window winder motors and four Mecanum wheels (Phillips, 2000). The aim of this project was to provide the platform with motion control that could be programmed to accommodate various robotic behaviours specified. With respect to the path following behaviour, two optical mice were attached to give positional feedback for closed-loop control and dead-reckoning for navigation and a Mitsubishi M16C/62 microcontroller was interfaced and programmed to implement robotic behaviours. A closed-loop control in Cartesian space was proposed to control x and y-movement and rotation motions of the robot.

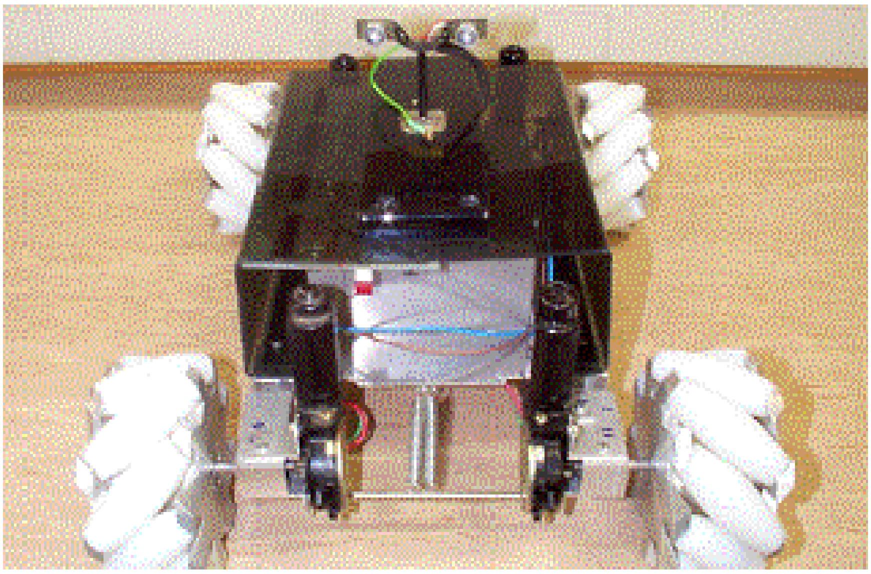
Source: Cutting Edge Robotics, ISBN 3-86611-038-3, pp. 784, ARS/plV, Germany, July 2005 Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

As this was a project incorporating mechanical, electrical and software development, a mechatronics design principle was applied. The different areas were developed synergistically thus allowing interactions between the disciplines to be viewed and managed. It also meant that all three core disciplines needed to be developed to a certain stage before any one area could be further worked on. Although it was physically possible to use other means to develop the core areas independently, a synergistic approach tends to be more efficient. Even though this parallel design approach was used, the areas of development shall be discussed in sections assuming that other sections have already been completed to a certain level and are referenced where necessary.

2. Robot Chassis

The original construction of the robot chassis (Phillips, 2000) included a form of “suspension”. The motor and wheel assembly at each corner of the chassis was mounted on a single shaft that allowed one degree-of-freedom between it and the chassis about the longitudinal axis. As shown in Figure 1(a) there was no mechanism designed to limit or control this degree-of-freedom and, once constructed, left a free moving joint. The two side assemblies at either end were subsequently linked via a piece of thin tubing in an attempt to limit and effectively prevent movement of the joints. A pair of springs had been added in an attempt to stiffen the setup further.

Some investigations were made into the system and it quickly became apparent that movement about the free axis was completely undesirable as it would allow the sides of the hub to foul the ground. Movement of this form would also affect the motion of the Mecanum wheel, as the sideways force vector generated would work directly against the “suspension”. When the wheel pivoted it would alter the current contact patch, thus altering the dynamics of the system. Early testing of the platform using the prototype driver board found this temporary solution unsuitable as it collapsed after only a short period of operation. The robot was to be used on level floors indoors, such as carpeted surfaces, so suspension considerations were therefore not warranted (Braunl, 1999). The motor and wheel assemblies were subsequently welded square to the chassis to provide a solid, robust, cheap and quick solution. Figure 1(b) shows the modified chassis.



(a) Original chassis with pivot system (Philipps, 2000)

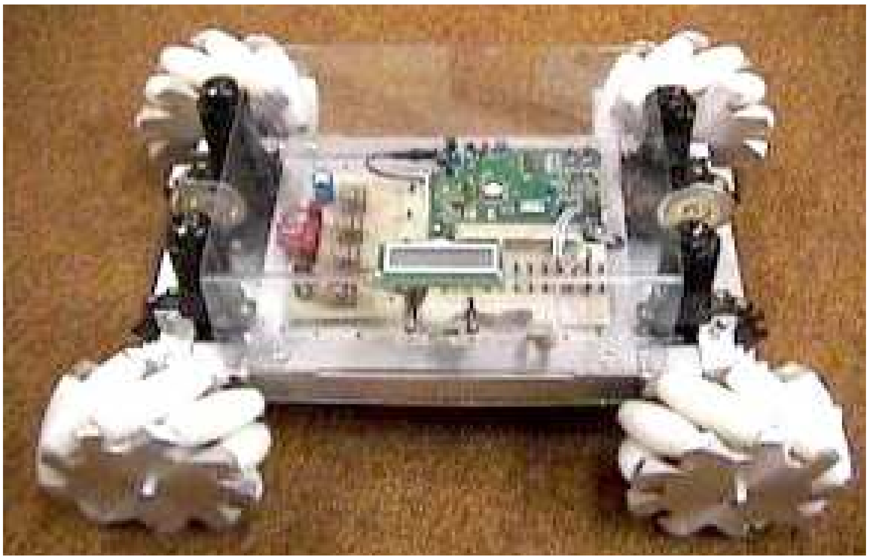

(b) The final robot after modifications

Figure 1. Mobile Platform with Four Mecanum Wheels

The other hardware modifications were as follows: the electronics were mounted on top of the chassis on stand-offs; a clear polycarbonate cover was formed to enclose the electronics and provide additional mounting space but still allow access to the circuitry and plugs via the open ends; a Liquid Crystal Display (LCD) was screwed up to the inside surface of the cover and power and program switches were positioned on the side of the cover; and a 18Ahr Sealed Lead-Acid was mounted beneath the robot platform. Major physical properties were measured and are given in Table 1.

Wheels		**Chassis**		**Overall**	
Diameter	150 mm	Length	463 mm	Length	543 mm
Width	75 mm	Width	260 mm	Width	460 mm
Mass	1.94 kg	Height	203 mm	Height	218 mm
Moment of Inertia	0.00546 kgm^2	Mass	12.15 kg	Mass	19.91 kg
Number of Rollers	9	Moment of Inertia	0.223 kgm^2		
Motor Torque	8.16 Nm				

Table 1. Physical Properties of Robot Platform

3. Motor Drive

The robot uses four used automotive electric window winder motors to drive the Mecanum wheels. These units are powered by a 12V DC motor which in turn is connected to a self-contained worm drive reduction. The output gear is connected via a rubber cushion drive. The reduction ensures that the motors provide considerable torque for driving the system. One disadvantage with these inexpensive used motors is that the high current demand of these motors requires a substantial portable power source.

The specifications developed for the motor driver board were:

(1) The circuit should be compatible with a single logic-level PWM input signal for the speed control of each wheel and a single logic-level input line for the direction of motor rotation for each wheel.

(2) The circuit should be able to operate with a high PWM carrier frequency (16 kHz or greater) to provide inaudible operation.
(3) The circuit would require four independent H-Bridge drivers.
(4) Each H-Bridge driver circuit must be capable of providing three amps continuous current at 12V DC.

The desired approach for the circuit design was to use dedicated motor driver Integrated Circuits (ICs) that were suitably rated. This would provide benefits of being compact, simple and being able to directly interface to a microcontroller. Investigation into the availability and pricing for dedicated ICs revealed that the options were too expensive to buy in the limited quantities required for this project. Instead the decision was made to construct a prototype quad MOSFET and relay H-Bridge driver circuit on Vero board to test the concept and enable further progression of the project. Each H-Bridge circuit consisted of one logic-level N-Channel 45Amp power MOSFET (PHB 45N03LT) and a Double-Pole Double-Throw (DPDT) relay. A signal transistor was used to interface the relay with logic-level signals. A Schottky diode was placed across the motor outputs to catch transient voltage spikes generated during the high speed switching of the MOSFET as a result of the inductive nature of motor coils. Figure 2 shows the schematic for a single H-Bridge. Four replicas of this circuit were built on the prototype board.

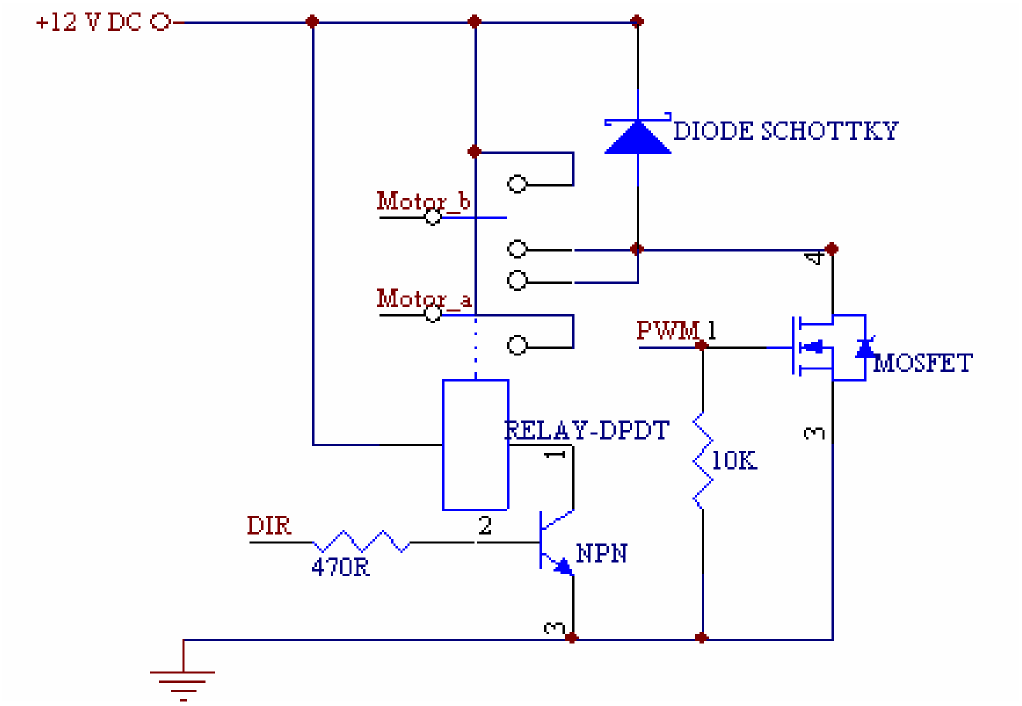


Figure 2. Single Prototype H-Bridge Schematic

Testing proved that this circuit design performed well and, with switching speeds of up to 20 kHz and full load applied to the motors, the MOSFETS did not overheat beyond their specification. However, one area of concern with the design was that relays, being a mechanical part, are subject to wear. This wear is greatly increased if they are switched at high current and high speed (in excess of 10Hz). This application differed though because the relay should only be switched when the motor is passing through zero speed and changing direction. Hence there should be no current flowing at the time the relay is switched. The circuit design was altered to include a charging jack and a voltage divider to provide battery status for use later in the project. The design was then laid out to create a Printed Circuit Board (PCB) for the circuit. The PCB was manufactured using available in-house equipment and implemented as a permanent solution.

4. Microcontroller

In order to give the existing robot any intelligent functionality some form of on-board processor was essential. Microcontrollers are ideally suited for such an application as they are compact, have many built-in hardware features such as timers and UARTS, have a significant number of digital I/O lines and have low power requirements. The essential microcontroller specification for this project was its ability to generate four independent PWM signals at frequencies greater than 15 kHz and with at least 8-bit resolution at these high frequencies. Other general requirements were; high speed operation to ensure environmental data could be processed at real-time, large RAM and ROM for complex algorithms, at least four 10-bit Analogue-To-Digital Converters (ADCs) to interface a sensory array and more than 30 digital I/O lines to interface peripherals such as an LCD display. A Mitsubishi M16C/62 was employed as it was available free of charge in the lab and also met the aforementioned requirements.

Connector	Pin Number	Function
1B/1C	VCC	Mouse 1, 2 & LCD +5V DC
2A	97	Battery voltage signal
5A	88	LCD data - DB0 (pin 7)
6C	87	LCD data - DB1 (pin 8)
6B	86	LCD data - DB2 (pin 9)
6A	85	LCD data - DB3 (pin 10)
7C	84	LCD data - DB4 (pin 11)
7B	83	LCD data - DB5 (pin 12)
7A	82	LCD data - DB6 (pin 13)
8C	81	LCD data - DB7 (pin 14)
8B	80	LCD function – RS (pin 4)
8A	79	LCD function – R/W (pin 5)
9C	78	LCD function – E (pin 6)
9A	76	Mouse 1 PS/2 bus – clock
10C	75	Mouse 2 PS/2 bus – clock
10B	74	Mouse 1 PS/2 bus – data
10A	73	Mouse 2 PS/2 bus – data
25C	28	Motor driver signal – PWM_1
25B	27	Motor driver signal – DIR_1
25A	26	Motor driver signal – PWM_2
26C	25	Motor driver signal – DIR_2
26B	24	Motor driver signal – PWM_3
26A	23	Motor driver signal – DIR_3
27C	22	Motor driver signal – PWM_4
27B	21	Motor driver signal – DIR_4
32B/C	GND	Mouse 1, 2 & LCD OV DC

Table 2. Pin Allocation for M16C/62 Microcontroller

The M16 is a 16 bit microcontroller and can be programmed using the high level 'C' language. The board contains many functions such as built-in timers, both hardware and software interrupts, A-D converters and 87 digital I/O lines. Its other major features are: 256KB ROM, 20KB RAM, two 7-segment LEDs, 16MHz main clock & 32KHz subclock, reset IC, switches, 5V regulator, RS232 driver chip, 96-pin DIN connector for application board interface. A pin allocation of the microcontroller interfacing for the robot is given in Table 2.

5. LCD Display for Robot Status

A 16 Character x 2 line LCD (Dick Smith Electronics – Z 4170), which is directly microcontroller compatible, was interfaced using an 8-bit bi-directional data bus. Three other Digital I/O lines were required for interfacing the Data / Instruction Select, Read / Write Select and Signal Enable lines of the display module. It was necessary to write a software driver for the display. The code was written as a module that could be easily used in any other programs. The software was derived from the data sheet provided with the display and example code found for similar displays.

The programs developed for this project required real-time communications and the use of an onboard UART and RS232 connector. As a result the PC software debugging suite could no longer be used. Instead the LCD was primarily used throughout the software development stage as a debugging tool. The final programs used the display to provide visual feedback to the user about the current status of the robot's operation, including position and battery status.

6. Optical Mice for Dead-Reckoning

Both navigation and path following require the robot to know its location. It was considered that a system using optical mice would provide greater flexibility when compared to other options (e.g., shaft encoder based dead reckoning, image based line following, beacon reference system).

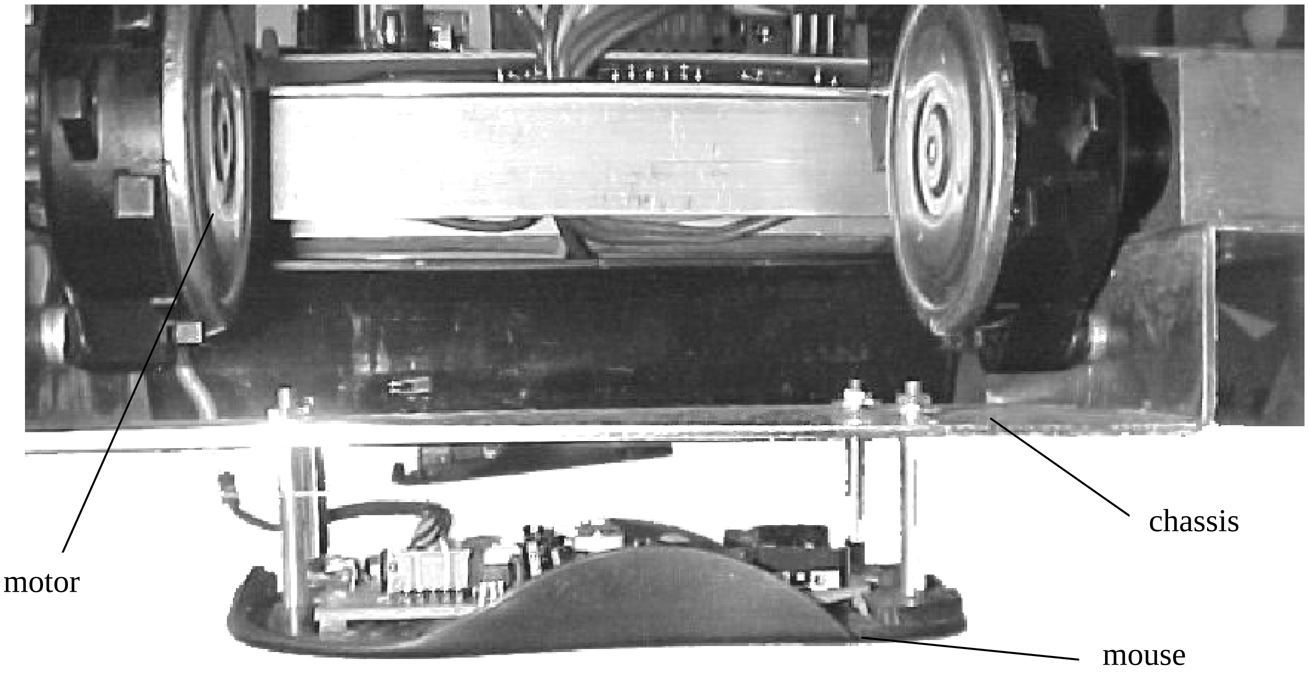


Figure 3. The Mounting of an Optical Mouse

An optical mouse has a tiny camera, a Digital Signal Processor (DSP) and microcontroller on-board which can provide a continuous data stream of x and y movements. Since an optical mouse can only measure translational movement not rotational movement, two optical mice were needed to sense motion in the robot's three degrees of freedom. By placing a mouse at either end of the robot looking down to the floor, as shown in Figure 3, the difference between the front and rear x-displacements is proportional to the angular rotation of the robot.

6.1 PS / 2 Protocol for Optical Mouse

Optical mice that were readily available could be interfaced via either the USB or PS/2 protocol and two A4Tech SWOP-35 optical mice were bought for NZ$38 each. However, since the M16C/62 microcontroller does not have hardware USB or PS/2 capabilities, a software PS/2 driver was derived and written from the specifications for the PS/2 mouse protocol.

The PS/2 protocol uses a two wire bi-directional serial bus. Both wires are open collector and can be pulled low by the host (microcontroller) or device (mouse). One wire is the clock signal which is provided by the device at anywhere between 10 and 20 kHz. The clock line can be held low by the host to inhibit the PS/2 bus. In this scenario the device will accumulate displacement until it overflows at which point data is lost. The second wire is the data signal which uses standard serial framing, one start bit, eight data bits, an odd parity bit and one stop bit (8-O-1). The Least Significant Bit (LSB) of the data is always sent first. The Clock line of the PS/2 bus is connected to a general purpose I/O line and the Data line is connected to one of the hardware interrupt lines of the M16C/62 microcontroller. A basic summary of PS/2 communications is as follows,

Device to Host communications (Figure 4)

- The device pulses the Clock line a total of 11 times, while transmitting the start bit, data bits, parity bit and stop bit on the Data line.
- The Host is expected to sample the Data line each time the Clock is low and the Device changes the state of the Data line while the Clock is high.

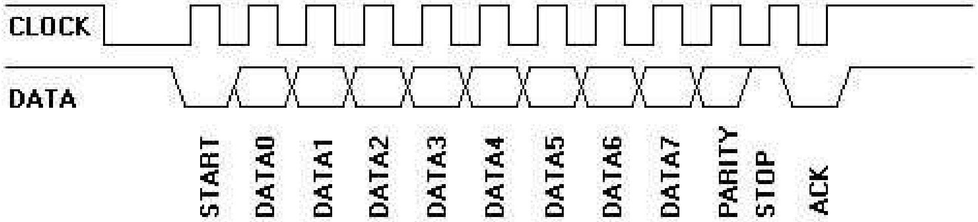


Figure 4. Device to Host Communications (Chapweske, 2002)

Host to Device communications (Figure 5)

- The Host signals its intent to transmit a command or argument byte by holding the Clock line low for at least 100µs, before pulling the Data line low and releasing the Clock line.
- The Device pulses the Clock line a total of 11 times to receive a byte of data. The Host is expected to change the state of the Data line while the Clock is low and the Device samples the Data line while the Clock is high.

- After the tenth Clock pulse, the Device checks for a valid stop bit (Data line high), and responds by pulling the Data line low and clocking one more time.

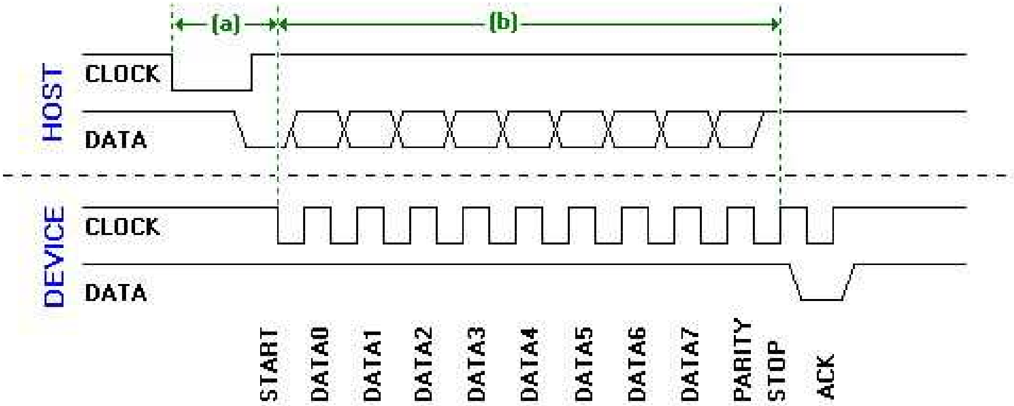


Figure 5. Host to Device Communications (Chapweske, 2002)

6.2 M16C / 62 Software Driver for Optical Mouse

The software driver written for the M16C/62 waits for the mouse to complete its power up self test and then issues a command to the mouse to enable data reporting (0xF4). By default, the mouse starts up in stream mode in order to continually generate x and y displacement data. This data is relative to the previous set of data whenever there is motion. However, the data is only transmitted on the PS/2 bus if data reporting is enabled. The acknowledge command (0xFA) is then expected to be returned, if it is not, the user is requested to reset the system. Once the mouse is initialised, the Clock line is held low by the microcontroller to inhibit the bus. At this point the main program is able to allow the Clock line to float high and read the data in when an interrupt is caused on the Data line. In default start up, the mouse operates in a standard PS/2 compatible mode. The data sent in this mode consists of a three byte packet. The first byte is a header, the second is the x data and the third the y data. Table 3 shows the detail of the standard PS/2 mouse data packet.

The optical mouse provides approximately 400 CPI (counts per inch) but testing proved that a value of 430 CPI, or 17 counts per mm was needed to properly calibrate the displacement data given by the mouse. The movement data is in 9-bit two’s complement, with the ninth bit located in the packet header.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Y overflow	X overflow	Y sign bit	X sign bit	Always 1	Middle Btn	Right Btn	Left Btn
Byte 2	X Movement							
Byte 3	Y Movement							

Table 3. Formatting of the PS/2 Mouse Data Packet

The mouse can also utilise the corresponding overflow bit in the header to increase its displacement accumulation buffer to 10-bit two's complement (± 512 counts).
This limits the displacement of the mouse between readings to *512/17 = 30mm per reading*. To enable the mouse to travel at 1m/sec it would need to be serviced every *30/1000 = 0.03 seconds*.

6.3 Testing

Preliminary testing was performed by using one mouse as an electronic ruler. The raw data received from the mouse was formatted according to the packet header. This data was accumulated for both the x and y-direction without scaling to maintain accuracy. The x and y-displacement accumulator values were subsequently scaled to millimetres to be displayed on the LCD panel. The parity of the data was also checked and an error was displayed on the LCD panel if a parity error occurred, to give an indication of the frequency of errors.
A mouse was temporarily attached to the front of the robot to test whether the performance of the measurement provided would be sufficient in the application. The tests showed that the optical mice provided better than 1% accuracy when the displaced measurement was compared to that given by a tape measure. Parity errors did occur but were very infrequent. The errors were most likely to occur if the mouse data cable was placed in close proximity to any one of the DC motors. To minimise the noise to which the data cables were subjected they were routed centrally down the chassis, while the DC motor power cables were routed through the side rails.
The final stage in fully implementing the optical mice on the robot was to create mounting brackets to hold the mice securely in the x and y-direction. Because the mice needed to be sitting flush with the ground it was necessary to implement some system to allow vertical movement. The mice were stripped down to the base and four shafts were mounted, one in each corner. These shafts were free to slide up and down in holes in the mounting bracket. Initially springs were used to provide some form of assisted return for the mice but testing proved that this added too much drag on the ground. Instead gravity proved to be sufficient to hold the mice to the ground.

7. Planned Path Motion

7.1 Open Loop Testing

With the mice installed it was possible to assess the open loop performance of the robot quantitatively, thus creating a benchmark for any further development. A standard planned path was designed that would provide the best assessment of the specifications. The path consisted of straight forwards, reverse, sideways and 45° diagonal lines. The 45° diagonal path was considered to be the most demanding of the robot as it is only driving with two wheels to achieve this direction. The standard testing path is shown in Figure 6. Due to the limited space available for testing, the maximum displacement from the centre of the course was 800mm. Table 4 gives the Cartesian coordinates for the path as used in the software.
The speed for forward and reverse movement was 152.88 mm/sec which gives 5.25 sec for 800 mm, the speed for sidewise movement was 152.28 mm/sec which gives 5.25 sec for 800 mm and the speed for diagonals was 167.19 mm/sec which gives 6.77 sec for 1130 mm.

	0	1	2	3	4	5	6	7	8	9
X	0	0	800	0	-800	-800	0	800	800	0
Y	0	800	800	0	-800	0	0	0	-800	0

Table 4. Planned Path Cartesian Coordinates

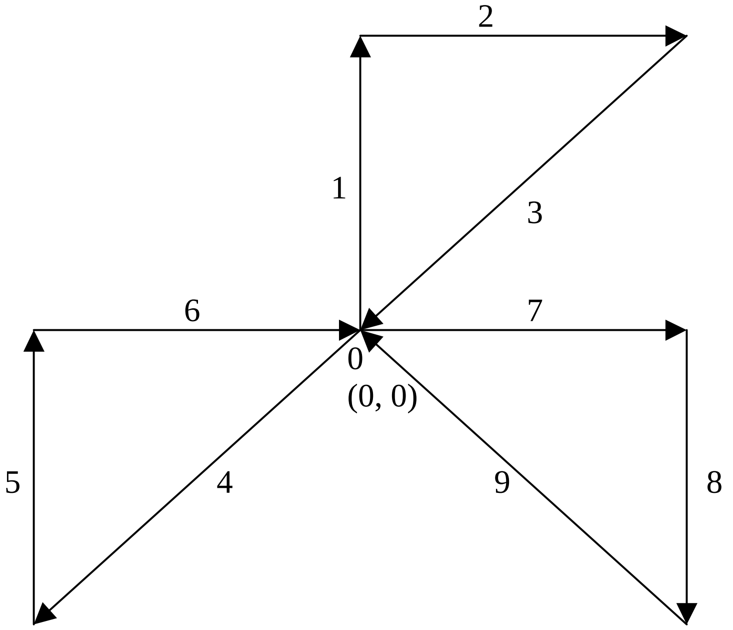


Figure 6. Planned Path for Testing

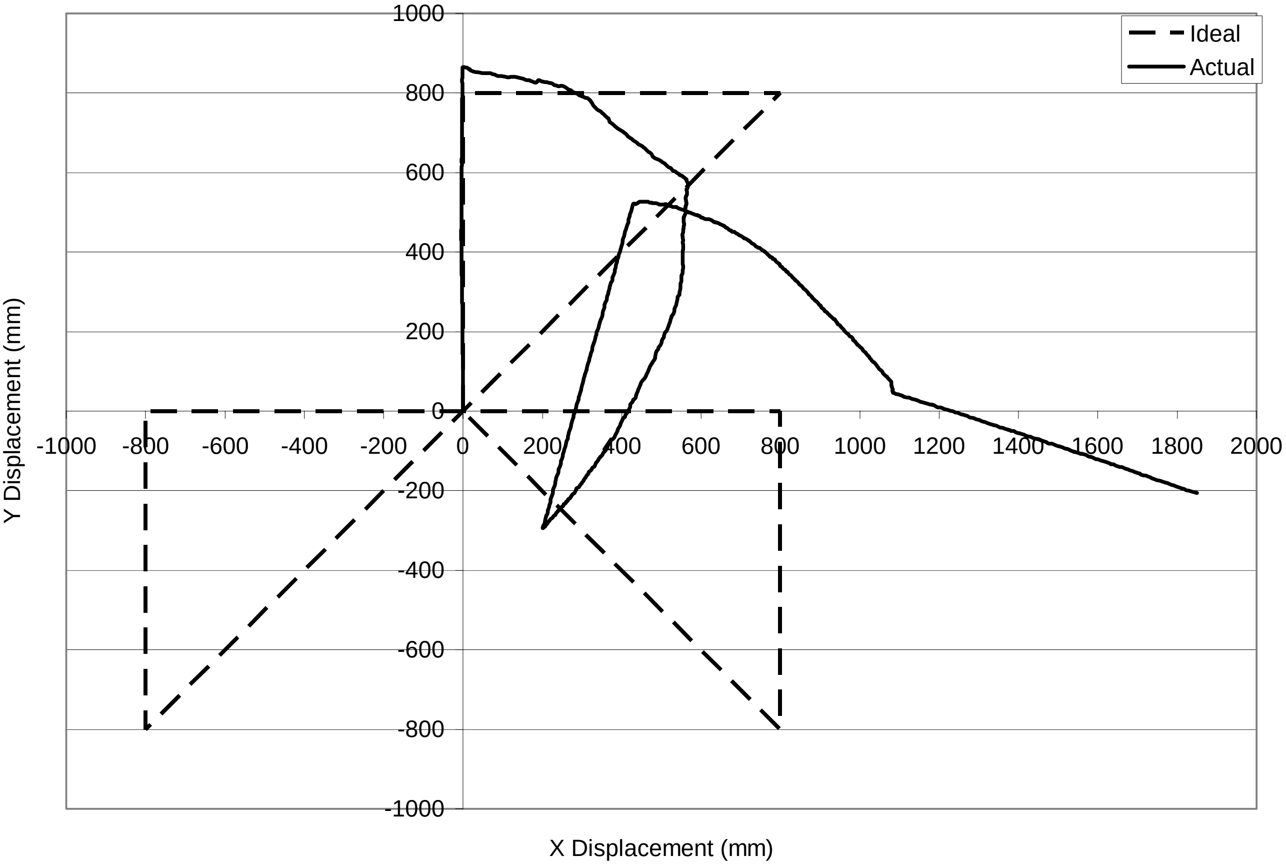


Figure 7. Open Loop Path Following Performance

7.2 Closed-Loop Control

In order to let the robot follow a planned path a closed loop control in Cartesian space was required. The simplest form of control for this multi-input, multi-output system is an array of Proportional Integral Derivative (PID) controllers as it requires no modelling of the system and is efficient to calculate for use in a fast acting real time application. The speed of the robot in the x and y-direction and rotation is controlled to bring the robot to the desired position otherwise known as a velocity servo controller. The problem was simplified by controlling the rotational position of the platform to a set-point of 0° and so simplifying the task of the microcontroller as the need to calculate trigonometry functions at real time, which is highly demanding on the processor, was alleviated. The set-points for the x, y and rotational speed controllers are proportional to the error between the desired x, y and rotational position and the actual positions. The desired value for the x and y position is calculated from the equation for a straight line ($y = mx + c$), between the previous and the current destination.

The three PID controllers were tuned using the Ziegler-Nichols Ultimate Gain method (Jobling, 2002) via three potentiometer voltage dividers connected to ADCs on the microcontroller. These were continuously scanned and scaled appropriately. Refer to Table 5 for the actual PID tuning constants. The software also streamed positional data through the UART to be logged and analysed by a PC. The path followed by the robot using closed loop control can be seen in Figure 8. A dramatic improvement can be seen over the performance of the open loop system.

	Proportional (P)	Integral (I)	Derivative (D)
X-Controller	3.38	0.154	1.02
Y-Controller	1.96	0.205	1.04
Rotational-Controller	2.72	0.343	1.21

Table 5. PID Tuning Constants

7.3 Results from Testing

Qualitative observations were made throughout the initial hardware development and modification stage of the project. These observations confirmed that modifications did increase the quality of the motion of the robot. The modifications to the pivot system had a successful result as they returned the chassis to a usable state that allowed further work and development to be done on the project.

Quantitative testing and analysis of the system was performed on the planned path motion application. The actual path travelled by the robot was compared to the desired path specified. Given in Table 6 are the results for this test calculated for both open loop control and closed loop control.

The specifications for the planned path motion were that the robot should not deviate more than ±1mm from the desired path for any forward or reverse movement, or ±5mm for any arbitrary angled movement within a two meter square area. The test data shows that neither of these specifications was met with open loop control. The implementation of closed loop control enabled the forward and reverse specification to be met. The diagonal path did prove to be difficult to control and although its performance lay outside the defined specification, it was a significant improvement over the open loop control.

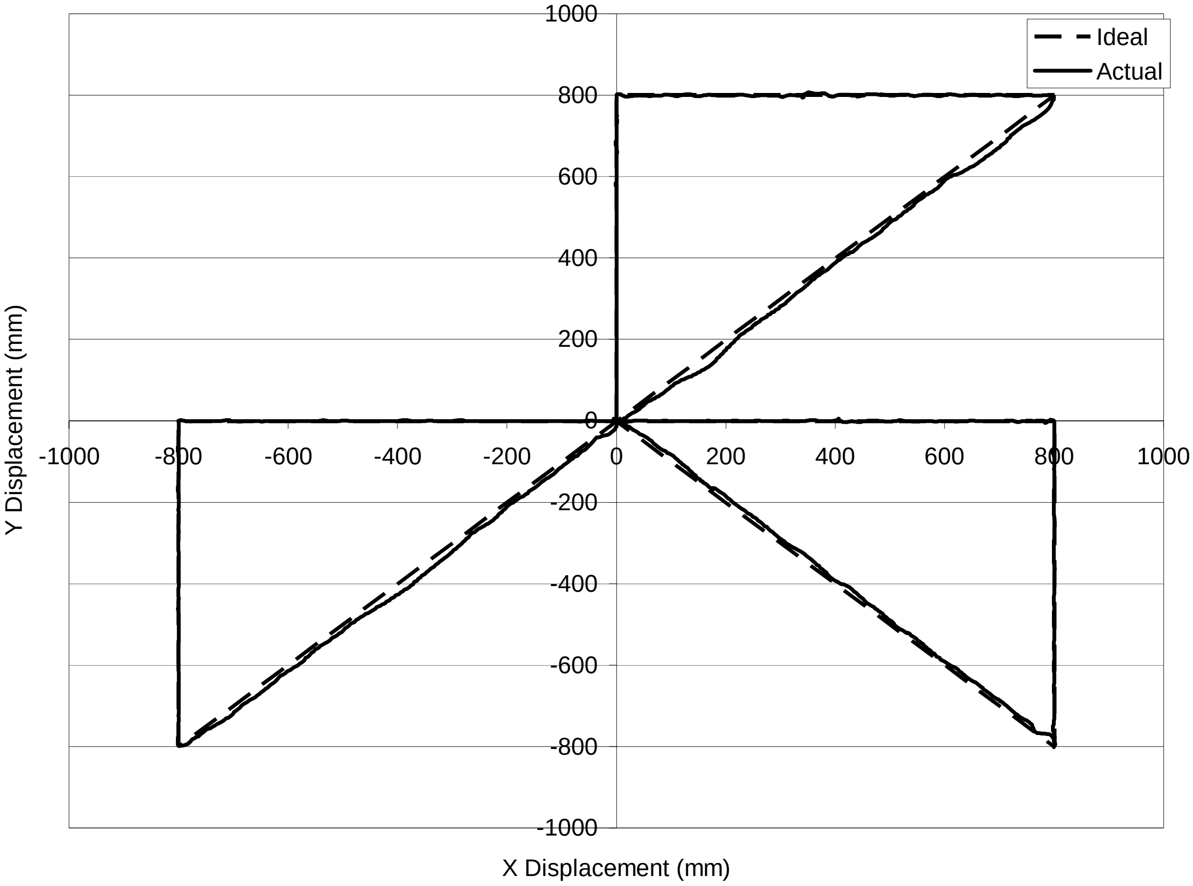


Figure 8. Closed Loop Path Following Performance

Open Loop Control				
	Forwards/Reverse	Diagonal (mm)	Left/Right	Rotation
Mean	-2.29	-47.72	415.74	-39.92
Standard Deviation	1.56	98.43	136.27	64.13
Minimum	-5.00	-239.00	6.00	-137.26
Maximum	0.00	65.00	496.00	27.69
Closed Loop Control				
Mean	0.04	9.56	-0.08	-0.07
Standard Deviation	0.65	8.49	1.43	0.72
Minimum	-2.00	-6.00	-6.00	-3.01
Maximum	1.00	36.00	7.00	7.53
Improvement	242%	1160%	9559%	8906%

Table 6. Results of Planned Path Testing

8. Conclusions

The motor driver circuit board designed for the drive of the Mecanum wheels met all the specifications given in Section 3. The limitation to its current handling capacity was the relays which are rated at five amps. The board has provided a cheap and compact interface between the microcontroller and the DC motors, fitting without trouble in the available space on the robot chassis. It is considered adequate to support further work done on the robot. Optical mice were successfully implemented on the system to provide positional measurement data to the microcontroller. They provided a cheap and accurate method of dead-reckoning for robot navigation, while making use of current technology in an application that is not currently common practice.

Wheel slip continued to be an issue with the Mecanum design. This problem was significantly improved by the use of closed loop control. PID controllers created a sturdy control design to investigate the platforms closed loop performance. Although the specifications set for the control task where not fully met, the performance of the platform was greatly improved.

The platform has been completed to a standard whereby further projects can use the robot to conduct advanced multivariable control investigations. The robot has been presented as a complete package, including a battery charger, in order to increase the ease of further research done on the project. If optimal system performance is required it is recommended that some modifications to the existing Mecanum wheels be made. This may necessitate new rollers being machined from a more appropriate material and, most likely, involve some form of rubber coating. To develop more sophisticated control of and telemetry from the platform, a wireless connection should be included. This would allow trigonometric, dynamic path and status calculations to be off-loaded from the microcontroller.

Acknowledgement

The work presented in this chapter is based on the research projects completed by J.G. Phillips and J.A. Cooney, Massey University, New Zealand, in 2000 and 2002, respectively.

9. References

Borenstein, J.; Everett, H.R.; Feng, L. (1996). Navigating Mobile Robots: Sensors and Techniques. A K Peters, Ltd, MA, USA.

Bräunl, T. (1999). Eye-Bot: a family of autonomous mobile robots. In: Proceedings of 6th International Conference on Neural Information Processing, pp. 645-649, November, 1999, Perth, Australia.

Chapweske, A. (2002). PS/2 mouse and keyboard protocol. Available from: http://panda.cs.ndsu.nodak.edu/~achapwes/ PICmicro/PS2/ps2.htm. accessed: October 2002.

Dickerson, S.L.; Lapin, B.D. (1991). Control of an omni-directional robotic vehicle with Mecanum wheels. In: National Telesystems Conference Proceedings, pp.323-328, March 26-27, 1991, Atlanta, USA.

Everett, H.R. (1995). Sensors for Mobile Robots: Theory and Application. A K Peters, Ltd, MA, USA.

Giachetti, A.; Campani, M.; Torre, V. (1998). The use of optical flow for road navigation. IEEE Transactions on Robotics and Automation. Vol.,14, No.,1, pp.34-48.

Jobling, C.P. (2002). Zeigler-Nichols tuning of PID compensators. Available from: http://eespectre.swan.ac.uk/~eechris/ ee306/docs/zeigler.doc. Accessed: October 2002.

Kraut, J. (2002). An Autonomous Navigational Mobile Robot. Bachelor of Science Thesis, Department of Electrical and Computer Engineering, University of Manitoba, Manitoba, Canada.

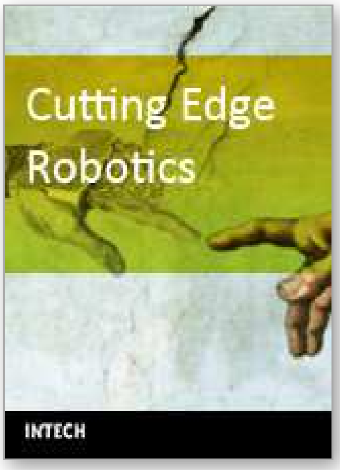
Lunze, J.; Schmid, C. (2002). Regelung einer mobilen Plattform via Internet. available from: http://www.esr.ruhr-uni-bochum.de/lehre/P1/P1_Fahrzeug_V6.pdf. accessed: October 2002.

Muir, P.F.; Neumann, C.P. (1990). Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In: Autonomous Robot Vehicles, Editors I.J. Coxy and G.T. Wilfong, pp.25-31, Springer-Verlag, Berlin.

Nagatani, K.; Tachibana, S. et al. (2000). Improvement of odometry for omnidirectional vehicle using optical flow information. In: Proceedings of IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp.468-473, Oct 30-Nov 5, 2000, Takamatsu, Japan.

Navy, USA. (2002). Omni-directional vehicles, Available from: http://robots.net/robomenu/996604291.html, Accessed: October 2002.

Phillips, J. G. (2000). Mechatronic Design and Construction of an Intelligent Mobile Robot for Educational Purposes. Master of Technology Thesis, Massey University, Palmerston North, New Zealand.



Cutting Edge Robotics

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-038-3
Hard cover, 784 pages
Publisher Pro Literatur Verlag, Germany

Published in print edition July, 2005

This book is the result of inspirations and contributions from many researchers worldwide. It presents a collection of wide range research results of robotics scientific community. Various aspects of current research in robotics area are explored and discussed. The book begins with researches in robot modelling & design, in which different approaches in kinematical, dynamical and other design issues of mobile robots are discussed. Second chapter deals with various sensor systems, but the major part of the chapter is devoted to robotic vision systems. Chapter III is devoted to robot navigation and presents different navigation architectures. The chapter IV is devoted to research on adaptive and learning systems in mobile robots area. The chapter V speaks about different application areas of multi-robot systems. Other emerging field is discussed in chapter VI - the human- robot interaction. Chapter VII gives a great tutorial on legged robot systems and one research overview on design of a humanoid robot.The different examples of service robots are showed in chapter VIII. Chapter IX is oriented to industrial robots, i.e. robot manipulators. Different mechatronic systems oriented on robotics are explored in the last chapter of the book.

How to reference

Peter Xu (2005). Mechatronics Design of a Mecanum Wheeled Mobile Robot, Cutting Edge Robotics, Vedran Kordic, Aleksandar Lazinica and Munir Merdan (Ed.), ISBN: 3-86611-038-3, InTech, Available from: http://www.intechopen.com/books/cutting_edge_robotics/mechatronics_design_of_a_mecanum_wheeled_mobile_robot

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821