Smooth Track-Keeping and Real Time Obstacle Detection Algorithm and Its PID Controller Implementation for an Automated Wheeled Line Following Robot

Article *in* Advanced Materials Research · February 2011

DOI: 10.4028/www.scientific.net/AMR.201-203.1966

S.M. Ferdous
Murdoch University
55 PUBLICATIONS **270** CITATIONS
SEE PROFILE

Enaiyat Ghani Ovy
University of Alberta
30 PUBLICATIONS **80** CITATIONS
SEE PROFILE

Md. Ashraful . Hoque
Islamic University of Technology
82 PUBLICATIONS **988** CITATIONS
SEE PROFILE

MIE10-112

Smooth Track-keeping and Real Time Obstacle Detection Algorithm with PID Controller for a Wheeled Line Following Robot

Mohammad Rokonuzzaman [1*], *S.M.Ferdous* [1], *Sayedus Salehin* [2], *S.M.A.A. Hasnine* [2], *Feroz Ahmed* [3]

[1] Department of Electrical and Electronic Engineering, Islamic University of Technology, Board Bazar, Gazipur-1704, BANGLADESH
[2] Department of Mechanical and Chemical Engineering, Islamic University of Technology, Board Bazar, Gazipur-1704, BANGLADESH
[3] Department of Electrical and Electronic Engineering, Khulna University of Engineering and Technology, BANGLADESH

ABSTRACT
Line following robots are extensively used in industries for smooth running of production and thus greater efficiency can be obtained if algorithm for smooth track keeping and obstacle avoidance can be implemented. This paper presents a simple and effective solution for path tracking problem for a wheeled mobile robot which can be used for material handling in industries. A PID controller has been used for controlling the robot which is capable of moving safely by smooth track-keeping in partially structured environment without any collision with static or moving objects. The purpose of the project is to build a mobile robot which will provide fast, smooth, accurate and safe movement in any given line or track. A straight or wavy line would be simple to follow whereas a T-junction, 90 degree bends, acute angle bends and grid junctions would be difficult to navigate through. This is due to the physical kinematics constraints which are limited to motor response, position and turning radius of the robot. A line sensor configuration has been proposed to improve the navigation reliability of the mobile robot which uses differential drive system. A dynamic obstacle detection algorithm has been developed for detecting obstacles which ensures the reliable and safe movement of the robot.

Keywords: Line following robot, Obstacle detection, PID controller, Microcontroller, sensors.

1. Introduction
Due to broad range of potential applications interest in mobile robot is continuously increased in last few years. Work has been done in the past to analyze various aspects of steering control in different situation for automobile steering or mobile robots. Some dealt with the Kinematics and dynamic involved [1], other have dealt with the controller design for a liner or nonlinear control [2]. Among different kind of mobiles robots differential drive wheel mobile robot have exhibit certain superior performance over other type of robots. A Differential drive system is a nonholonomic system [3]. Work has been done to model and analyze the nonholonomic differential drive system [4]. Close loop control and trajectory control model is proposed [5]. In this paper we have designed a control system for real world application and the designed control system is implemented using microcontroller based differential drive system. Many authors work had been studied [6, 8, 9, and 10] for the design of the smooth track keeping mobile robot which is also capable of detecting obstacle. The robot is designed to perform the following task:

- Follow the line accurately and smoothly.
- Stopping when derailed from operation line.
- Stopping and flashing a light when an obstacle is encountered.
- Start following line automatically when the obstacle is removed.

A microcontroller is used for controlling the robot according input sensor values. A number of Infrared sensors are used for detecting and following lines. Also for detecting the obstacle in front of the robot is determined using another infrared sensor.

2. System Development
2.1 Motor Control: The mobility of the line following robot is obtained with two DC geared motors which are attached to the wheels of the car. Control for the two motors in the system is carried out by using the L293D integrated circuit H Bridge. The driving signals are generated by the microcontroller which produces appropriate PWM according the position of the robot on the line

2.2 Sensor Placement
2.2.1. Sensor arrangement for line detection
Six infrared sensors are used and the arrangement is shown in Fig. 1 which is placed underneath the robot. This arrangement is used for accurate detection of line for different kind of line curvature and junctions.

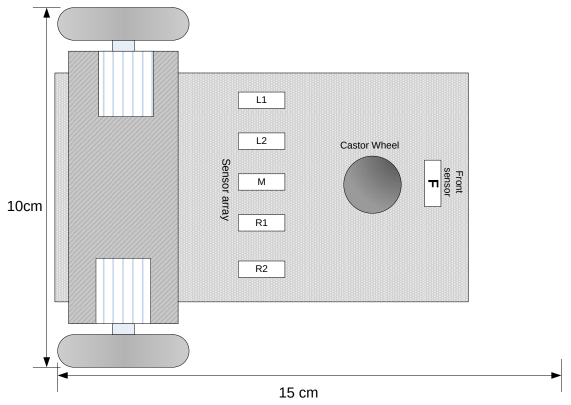

Fig.1: Robot chases with line following sensors.

* Corresponding author. Tel.: +88-01719338349
E-mail address: rokon.iut@gmail.com

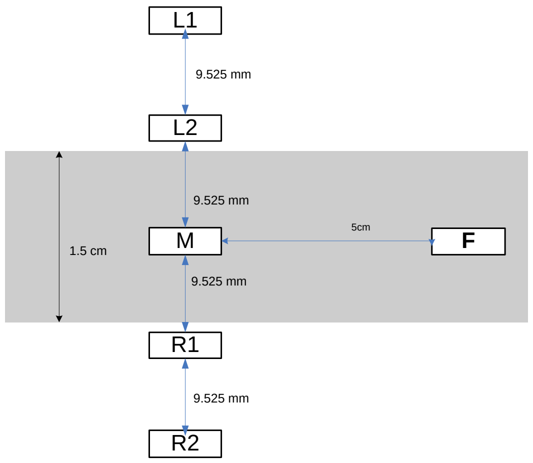

Fig.2: Distance distribution of Sensors within the layout platform

The sensors used as shown in Fig. 2 used are all analog sensors which provide analog output values and after converting these analog values to binary using microcontroller's ADC, a range of output 0-1024 is achieved (distance should be same all the time between the sensors and the surface). These output values are used to differentiate between the line and outer surface; these values are tabulated in Table 1.

Table1: Sensor output data (digital) for different surface

Color of the surface	Range of Sensor Output
Black	880-900
Red	120-130
Yellow	30-35
White	20-25

When the sensor is on the black line (value within the range of 650- 950) we consider the sensor output is 1 and when the sensor is outside the line (Values less than 650) 0 is considered as the output of the sensor.

2.1.2 Sensor arrangement for obstacle detection

An infrared sensor is used in front of the robot for the detection of the obstacle in front of the robot. The sensor arrangement of the sensor for obstacle detection is shown in the Fig. 3:

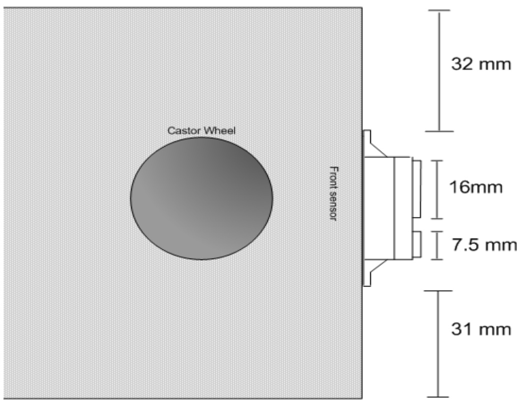

Fig.3: Sensor placement for Obstacle Detection

3. Line drawing Analysis:

3.1 Straight line

When the sensors F and M is on the line and L1, L2 and R1, R2 are out of the line means the robot is on the straight line so both the motor will rotate forward and the robot will move forward in its highest assigned speed.

3.2 Curve lines

A curve line is detected when the sensor F is out of the line. However in some cases the sensor may go outside the line due to overshoot of the robot which is discussed in the following sections. For both cases the robot will follow the same algorithm. Sensor outputs for all possible positions and the corresponding response of the robot are tabulated in Table 2.

Table 2: Sensor outputs for all possible position and corresponding action to be taken.

01110	Centered over line, increase speed for straight runs.
01111	Slightly to the right of the center of the line, slight correction to the left
00111	Right of the center of the line, steer left
00011	Near the right edge of the line, steer left
00001	Almost off the line, steer hard left and reduce speed
11110	Slightly to the left of the center of - the line, slight correction to the right
11100	Left of the center of the line, steer right
11000	Near the left edge of the line, steer right
10000	Almost off the line, steer hard right and reduce speed
00000	Lost line from overshoot or break in line
11111	Line intersection

While a curve line is been followed, the front sensor is out of the line and the robot follows the above mentioned algorithm.

4. PID controller Design

PID controller is the mathematically based routine that processes the sensor data and control the position of the robot to keep it on the course. This PID controller is used to control the robot with quick response time and to minimize the overshoot as much as possible.

A robot without the controller would oscillate a lot about the line, wasting time and battery power. Using the PID controller the robot will follow the line smoothly keeping its center always above the line. In straight line the robot will gradually stabilize itself and this will enable the robot to follow the line faster and more efficiently.
A standard PID controller action can be expressed by-

$$u(t) = k_p e + k_i \int e dt + k_d \frac{de}{dt}$$

Here,
e= error signal
k_p = proportional gain
k_i = integral gain
k_d = derivative gain
This can be rewritten as –

$$u(t) = k_p [e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt}]$$

Where, $k_p = \frac{1}{x_\rho}$; x_ρ = proportional band

$T_i = \frac{K_i}{x_\rho}$ = integral action time

$T_d = k_d \ x_\rho$ = derivative action time

The transfer can be obtained as-

$$G_c(s) = \frac{Y(s)}{E(s)} = K_p (1 + \frac{1}{T_i s} + T_d s)$$

Where, Y(s) is the input signal.

Proportional controller measures how far the robot is away from the line, it produces bulk of corrective effort. The more granular sensor data is, the more accurately robot's position can measured over the line.
Integral control measures the error over time. It eliminates the proportional offset (steady state error). The value increases while robot is not centered over the line.
Derivative controller measures the path the robot moving apart from the center. Any change in control deviation will result in output proportional to speed of which is the deviation is changing which will eventually minimize overshoot.

Sensing line and maneuvering the robot to stay on course, while constantly correcting wrong moves using feedback mechanism forms a simple yet effective closed loop system. The controller first calculates the correct position and calculates the error (if any). Then it will command the motor to take a hard turn if the error is high or small turn whenever error is lower. So the magnitude of turn will be proportional to the error. This is the due to the proportional control applied in the system. If the error does not decreases or decreases slowly, the controller will increase the magnitude of the turn further and further over time till the robot centers itself on the line. This is due to the effect of integral control. In the process of centering over line the robot may overshoot from the target position and move to the other side. This robot may keep oscillating about the line. To reduce the oscillating effect over time derivative control is used. The designed controller can be modeled with the following block diagram as shown in Fig. 4.

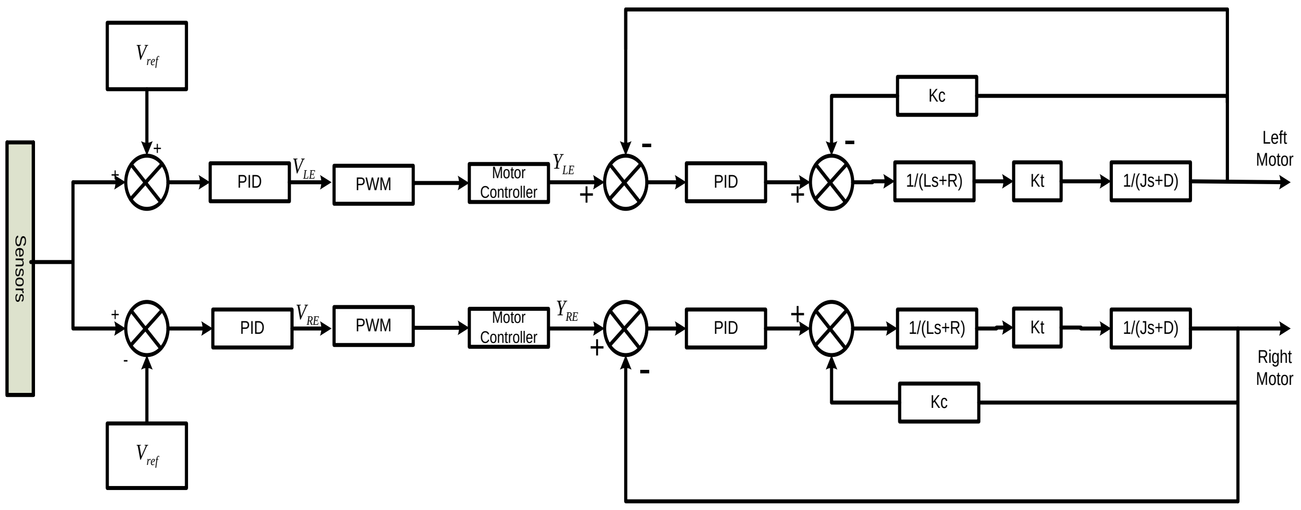

Fig. 4: Block diagram of the control system

5. Practical Implementation:

PID is actually a differential equation solved in the frequency domain. However, with some assumptions and simplifications the math becomes very simple and lends itself to rational real world explanations. Now for implementing the PID formula in a microcontroller the PID formula can be simplifies as follows using trapezoidal rule [8].

$$Y_n = Y_{n-1} + K_p(e_n - e_{n-1}) + k_i(e_n - e_{n-1})/2 + K_d(e_n - 2e_{n-1} + e_{n-2})$$

Here, Y_n = current motor response, Y_{n-1} = previous motor response, k_p = proportional parameter factor, e_n = current error, e_{n-1} = previous error1, k_i = Integral parameter factor, k_d = Derivative parameter factor, e_{n-2} = Previous error 2

In this part the control system is implemented with the microcontroller. We have considered the possibility of two types of error. As we always wanted our robot to be on the desired position (completely on the line). We determine the position of the robot on the line based on the output of the. When the robot is on straight line the output of the five sensor array is 00100. This is our target position. When is deviated from the line the output pattern of the sensor array will change. Error value assigned by determines how far the robot deviated from the line. It is desired to obtain a linear relationship between line position and sensor output.

Table3: Error value depending on the robot position

Sensor output	Error value
00001	4
00011	3
00010	2
00110	1
00100	0
01100	-1
01000	-2
11000	-3
10000	-4
00000	5 or -5 depending on previous value.

To control the speed of the motor, it has to be varied by PWM technique for that respective motor depending on the error (that is left turn or right turn).PWM is done by changing the duty cycle of the input voltage to motor. As we intended to implement a linear relationship between the sensor output and robot speed two equations are derived from the algorithm to control the speed of the motor and implemented using the microcontroller.
From figure 4:
Reference Value = 5

V_{LE} = V_{ref} + e

V_{RE} = V_{ref} - e

For left motor:

Duty cycle (PWM) = 20$\times V_{LE}$; $0 \le V_{LE} < 5$

Duty cycle (PWM) = 100 ; $V_{LE} \ge 5$

For right motor,

Duty cycle (PWM) = 20$\times V_{RE}$; $0 \le V_{RE} < 5$

Duty cycle (PWM) = 100 ; $V_{RE} \ge 5$

Where Y_{LE} and Y_{RE} in Fig. 4 are the motor responses from the PID controller for left and right motor respectively.

6. Simulation and PID controller tuning

For tuning the controller we have used the Ziegler-Nichols ultimate cycle method as it is more convenient than process reaction curve method. Then the tuning is verified by putting different values of K_p, K_I and K_d by trial and error method and simulating in MATLAB. First all the values of the gain are set to minimum and using ultimate cycle method we found the Critical value of proportional constant K_{pc} and from the time domain response curve the periodic time of oscillation T_c is measured. By using the Table of ultimate cycle criteria for PID controller value of K_p, T_I and T_d is determined where,

$K_I = 1/T_I$; $K_d = T_d$;

Control Mode	K_p	T_I	T_d
P	0.5 K_{pc}	-----	-----
PI	0.45K_{pc}	T_c/1.2	-----
PID	0.6K_{pc}	T_c/2.0	T_c/8

The time domain response curve is simulated in SIMULINK. Every time different values of K_p is used and the response is observed. Finally a critical value (to be put) is found for which the system is oscillatory or marginally stable (poles on the imaginary axis) and the value of the gain factors are determined accordingly following the procedure. In the software the same value is used and the robot response is observed. The same is repeated with several other values of gain factors and the value obtained by the Ziegler-Nichols method is proved to be the optimum.

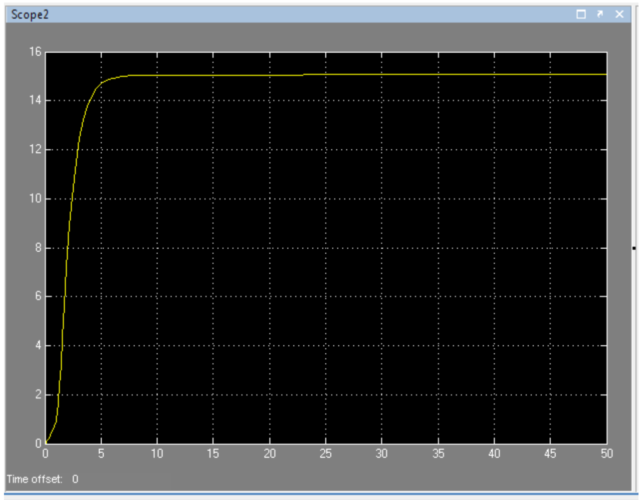

Fig. 5: Response curve of the robot control system (error simulated as triangular wave.

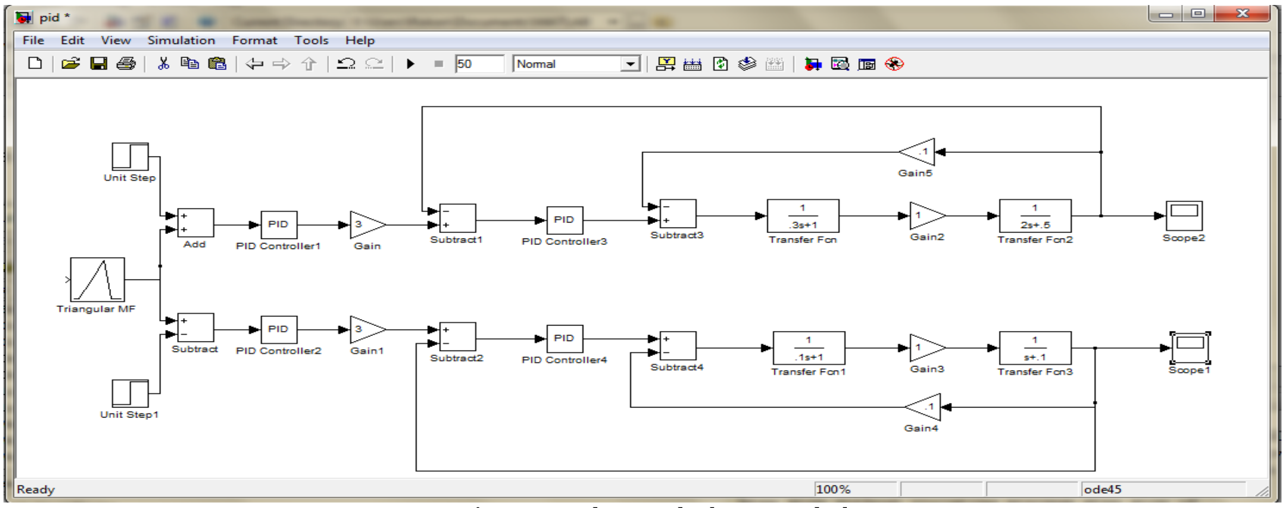

Fig. 6: Simulation Block in Simulink

7. Obstacle Detection

In obstacle detection algorithm, whenever the robot detects an obstacle by measuring the analog output voltage of the IR Sensor which exceeds the threshold limit, it stops in his track. In this purpose sharp GP2D12 IR sensor is used which gives analog output depends on the distance between the obstacle and the robot. When there is no obstacle if provides very low output voltage and the closer the obstacle the higher the output voltage of the sensor is.

This part is for the safety of the robot to avoid any collision if there any obstacle in the track by accident. The microcontroller continuously looks for the IR output whether the obstacle is being removed from the track or not. If the obstacle is not removed within 30 seconds it provides an indication by flashing a LED.

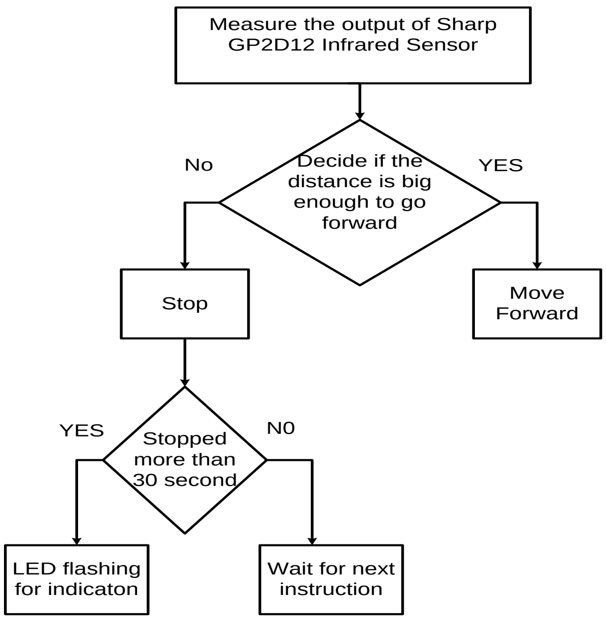
Measure the output of Sharp GP2D12 Infrared Sensor
Decide if the distance is big enough to go forward
No
YES
Stop
Move Forward
Stopped more than 30 second
YES
NO
LED flashing for indicaton
Wait for next instruction

Fig. 7: Flow chart for obstacle detection

8. Conclusion:

The principle complicacy of the design is to tune the PID controller and to get an optimum value for the gain factors for an optimum fast dynamic response of the robot. Interestingly it has been observed that this values changes time to time in different conditions. As a result no fixed value can be set for the robot PID controller. By storing the value of K_p, K_I and K_d in the EEPROM of the microcontroller may solve this problem so that the controller does not need to be tuned each and every time. Using this approach the tuning problem may be overcome or tuning of the controller will be more flexible. We intend to implement it in future. Our future work also includes increasing the number of sensors to get a more fast response and detect the curvature of the line with more precision and accuracy. On the hand it does not have the provision of obstacle detection and avoidance. This can be added with the robot by adding some extra sensors and developing an algorithm for implementation in the software. This extra feature can make the robot more versatile and the robot it self may be used for some practical applications in the field of automation. If it is possible to develop such an algorithm that will enable the robot to map the line it is following then the best possible response can be achieved by the robot. This type of algorithm may not be possible to implement without image processing and at the same time cost will be increased due to addition of camera and other related opto-electronic devices. We shall try to develop such algorithm with conventional sensors and implement it.

REFERENCES

[1] Tagawa, Y. Ogata, H. Morita, K. Nagai, M. Mori, H. Robust active steering system taking account of nonlinear dynamics Vehicle System Dynamics. v 25 n Suppl 1996. p(668-681).

* Corresponding author. Tel.: +88-01719338349
E-mail address: rokon.iut@gmail.com

[2] Hattori, A. Kurami, K. Yamada, K. Ooba, K. Ueki, S. Nakano, E., Control system foran autonomous driving vehicle Heavy Vehicle Systems. v 1 n 1 1993. p 99-113

[3] C. I. Connolly and R. A. Grupen, Nonholonomic path planning using harmonic functions," Tech. Rep.
94-50, Computer Science Department, University of Massachusetts, May 1994.

[4] Theoretical aspects in wheeled mobile robot control, M. NiţulescuUniversity of Craiova, Romania,Faculty of Automation, Computers and Electronics,Automation, IEEE International Conference on Quality and Testing, Robotics, 2008.

[5] Feedback control design of differential-drive wheeled mobile robots, Shouling He; Dept. of Electr. & Comput. Eng., Pennsylvania State Univ. 12th International Conference on advanced Robotics, 2005.

[6] A Smooth Path Tracking Algorithm for Wheeled Mobile Robots with Dynamic Constraints,K. C. KOH Division of Mechanical and Control Engineering, Sun Moon University, 100 Kalsanri, Asanshi,,H. S. CHO Department of Mechanical Engineering, KAIST, Journal of Intelligent and Robotic Systems 24: 367–385, 1999.
[7] P.F. Muir and C.P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots,"
Journal of Robotic Systems, 4(2), 1987, pp. 281-340.
[8] Thomas Braunl, Embedded Robotics, Second Edition Springer-Verlag 2006:
[9] B. Andrea, and G. Bastin, “Modeling and control of non-holonomic
wheeled mobile robots”, Proc. of IEEE International Conference on Robotics
and Automation, Sacramento, SUA, p. 1130, 1991.
[10] R. De Santis, “Modelling and path tracking control of a mobile wheeled robot with a differential drive”, *Robotica*, 13, Part 4, p. 401, 1995.