

An Educational Velocity Controlled Robot Vehicle

Mr. Ezzaldeen Edwan *

ABSTRACT

In this paper we describe a novel velocity controlled robot hierarchy implemented on a modular autonomous platform. The velocity controlled robot is achieved using traditional PID controllers. This vehicle will be a testbed for educational purposes and research in cooperative multi-vehicle systems. In order to be autonomous, it must have localization techniques and navigation control strategies.

Keywords: robot, velocity controlled, autonomous platform.

* Palestine Technical College, Deir Elbalah, Gaza Strip, Palestine, edwan@ieee.org

INTRODUCTION:

The latest advances in information and sensors technology present unprecedented opportunities in using them in applied aspects. We now have the necessary resources to develop modern robots. Developing unmanned vehicles for unknown, unstructured scenarios is an intensive area of current research and until today, the autonomous navigation is an open challenge for mobile robots developers and researchers.

A. Why We Need a Modular Mobile Robot?

Nowadays there is an increasing demand on modular mobile platforms for research and many practical applications. The prices of these platforms are very high which limit many researchers from doing experimental work on mobile robots. Another restriction is that most of the existing platforms are limited to indoor use only. In our laboratory, we have developed a modular platform with off-the-shelf components. The designed vehicle is low cost and can be used for indoors and outdoors operation.

Mobile Robot Modeling:

In our analysis, we will consider the kinematic model which deals only with nonholonomic constraints that result from rolling without slipping between wheels and ground.

A. Kinematic Model of Disc Wheel (Unicycle)

Considering the car as a rolling disc wheel (unicycle) and assuming pure rolling and no slipping, the kinematic model explained in [1] is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

Where (x, y) is the Cartesian position and θ is the orientation with respect to the positive x axis, v is the linear velocity of the disc wheel and ω is the angular velocity of the disc wheel.

B. The Car-Like Model (Double Steering)

In double steering we have both the front wheels and rear wheels to be steerable. This configuration gives us a smaller radius of curvature compared to single steering. With this feature we can have faster convergence to the desired orientation of car body. Double steering does not affect the degrees of freedom for mobile robot. Fig. 1 shows a model for double steering. In this model, we will assume both steering angles are equal and opposite in direction to simplify the computing of the kinematic

model. Taking (x, y) as the Cartesian coordinates of the point in the center of the car and applying rolling without slipping to both wheels we get

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 2 \frac{\tan \phi}{l} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (2)$$

Where v_1 and v_2 are the driving and steering velocity inputs, respectively.

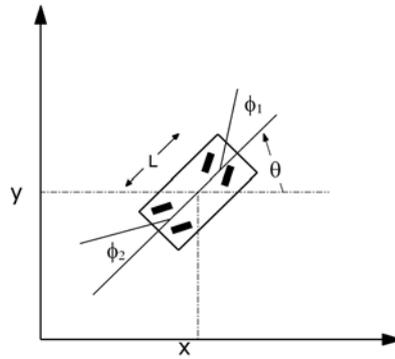


Figure (1): Global coordinate system for the car (double steering)

The maximum steering angle places a limitation on the maximum angular velocity that can be achieved by the robot. The minimum radius of curvature occurs at maximum steering angle.

$$R_{\min} \approx \frac{1}{\alpha \phi_{\max}} \quad (3)$$

The parameter α is found experimentally. The angular velocity is constrained by

$$-\omega_{\max} \leq \omega \leq \omega_{\max} \quad (4)$$

Where ω_{\max} is computed from the following equation:

$$\omega_{\max} = \frac{v}{R_{\min}} \quad (5)$$

Therefore, we can substitute ω_{\max} in equation (4), and then it can be rewritten in the following new formula

$$-\frac{v}{R_{\min}} \leq \omega \leq \frac{v}{R_{\min}} \quad (6)$$

From the above equation, we can find the limits by substituting in the minimum radius of curvature. Fig. 2 illustrates the angular speed in its contradictory to the linear speed relationship

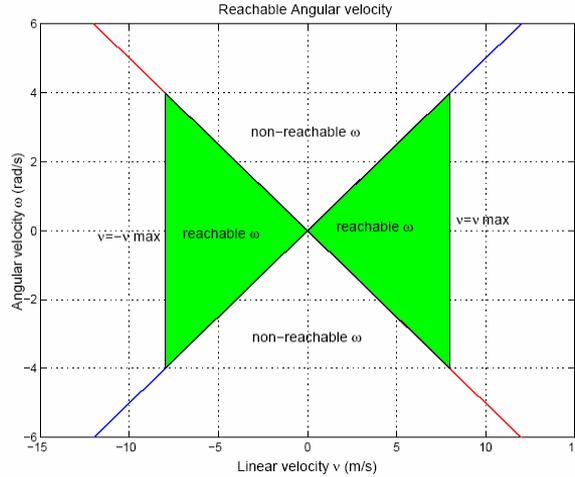


Figure (2): Angular speed vs. linear speed relation

The Unicycle versus Car-Like Model:

Considering the first three states (x, y, θ) in equation (2), the car-like model reduces to the unicycle model. Using the approximation of $\tan \phi \cong \phi$ for small angle ϕ , we can express ω as

$$\omega = 2 \frac{\phi}{l} v_1 \quad (7)$$

Hardware Description:

A. The Experimental Testbed:

The MARHES Laboratory X-treme robots are based upon TXT- 1, a commercially available radio control truck from Tamiya Inc., with significant modifications [2]. An on-board notebook computer provides the computational power for signal/image processing and motion control. Also, a PCMCIA Multi-function I/O card from National Instruments was used for interfacing the computer with a suite of analog and digital sensors. The suite

of sensors includes IR distance sensors odometer, receiver, compass and stereo vision camera.



Figure (3): The MARHES laboratory experimental testbed.

B. Dead Reckoning Sensors:

Angular rotation of the wheel is measured by a rotary optical encoders attached to the wheel shaft. We used two optical encoders placed inside each of the front wheels of the robot. Using two quadrature encoders provide us with both linear and angular displacements. The data output of each quadrature encoder is sent to one of the two counters in the data acquisition card. One direction signal from either of the quadrature encoders is adequate to tell about the direction of rotation in the 4-wheel robot. The counter counts the rising edges for a specified period of time. The counter from the NI-DAQ card can be reset at any instant and is automatically reset once it reaches the maximum number counts. For this purpose, we should use the calibration procedure described in [3].

Velocity Controlled Robot:

Working with a kinematic model is simple and makes the realization of the controller possible. The usual inputs for a 4 wheel mobile robot are the steering angle and speed. Dealing with these two inputs might increase the difficulty of controlling the robot since many control laws are expressed in terms of linear and angular velocity. One can think that a higher level controller (planner) generates the desired velocities and a lower level controller deals with the car dynamics (mass, inertia, etc.). As a result, we need a transformation that transforms the input commands of linear velocity (v) and angular velocity (ω) to motor speed and steering angle. Most applications for controlling mobile robots require accurate matching

An Educational Velocity Controlled Robot Vehicle

between commanded velocity and the actual velocity. This property can not be achieved using an open loop controller. Hence there is a need for a closed loop controller to ensure the convergence of the actual velocity to the commanded velocity. PID controllers have been used effectively for many years in industry. Nise [4] has a good discussion of adjusting the PID gains. Fig. 5 shows a block diagram of the velocity controlled robot.

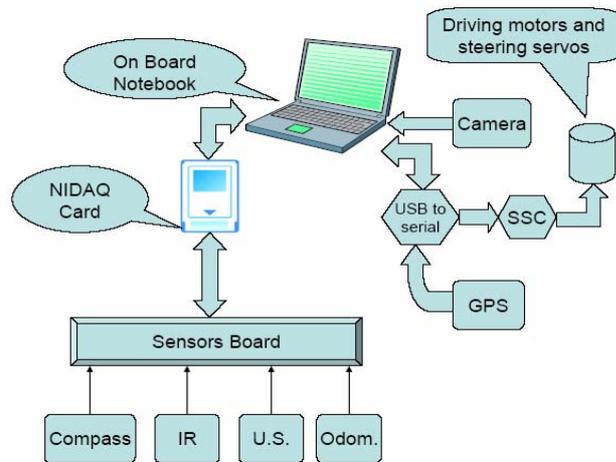


Figure (4): Block diagram of robot hardware.

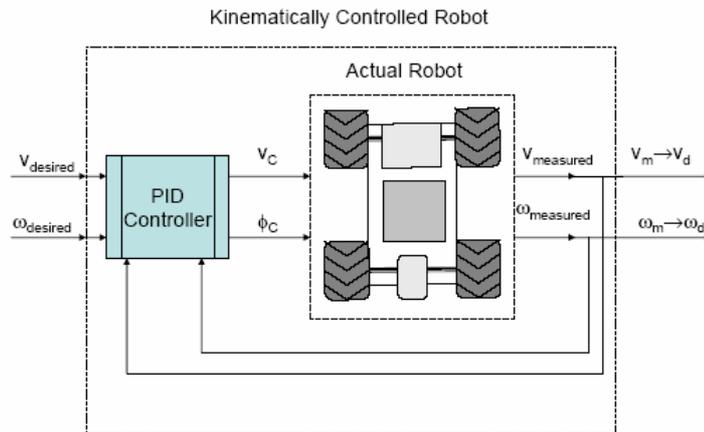


Figure (5): Velocity controlled robot.

A. PID Linear Speed Controller:

Many control applications assumes a velocity controlled thus maintaining the actual speed as the desired one is important. For a non-inclined surface the speed can remain constant, but if the surface becomes inclined this condition is violated. Therefore, we implement a digital PID speed controller based on the installed optical quadrature encoder sensor. The speed is calculated by knowing the difference between two encoder readings and the time elapsed between the two readings. In our subroutines, a sampling rate of 200 msec is implemented using the timing function. After some experiments in measuring the speed using this sampling rate, it was found that a minimum speed of 0.10 m/s can be measured with acceptable accuracy and as the speed increases the accuracy improves. The robot linear velocity is actuated by two DC motors. We approximate the system as a first order system with the transfer function

$$G(s) = \frac{K}{1 + \tau s} \quad (8)$$

Where K is a constant and τ is the time constant of the system response. The time constant depends upon the robot's environment. A simple standard digital PID controller has the general form:

$$u = k_p e + k_i \int edt + k_d \dot{e} \quad (9)$$

The gains were tuned experimentally to obtain the desired overall response. In discrete time intervals the controller has the given form:

$$u_k = k_p e_k + T_s k_i \sum e + \frac{k_d}{T_s} (e_k - e_{k-1}) \quad (10)$$

Where T_s is the sampling period.

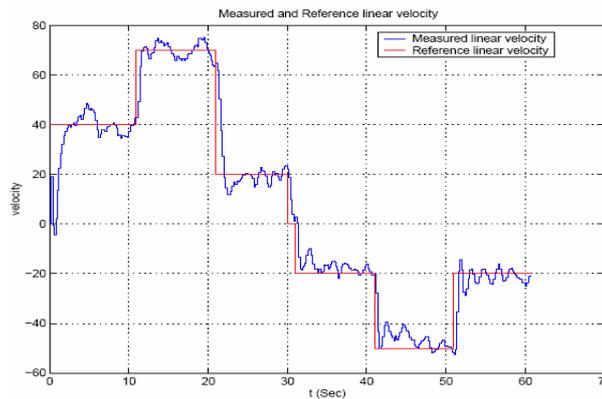


Figure (6): Measured and reference linear velocity

B. PID Angular Speed Controller:

The basic idea behind this controller is, if the linear velocity is assumed to be constant then the steering angle will control the angular velocity. The relation between v and ω is

$$\omega = \frac{v}{R} \quad (11)$$

Here, R is the signed distance from the ICC (instantaneous center of curvature) to the mid point between the two front wheels. From equation (3), R is inversely proportional to steering angle. Thus we can use the same form of the previous PID controller in equation (10) to regulate the steering commands instead. One important parameter to consider is the sampling time. The sampling time is limited by the steering servo response time (i.e., the sampling time can not be smaller than steering servo response time). The response time of the servo is constant at no load but it varies according to the wheel-ground friction. The design of the PID controller will be for a nominal value of v because ω is coupled with v as in equation (7). The angular speed controller assumes v is constant. This problem is eliminated by having a dynamic PID gains. In other words, the PID gains will vary based on the value of desired linear velocity. Fig (6) shows the measured linear velocity with a constant reference linear velocity.

Localization:

In this section, we discuss the effectiveness of our designed robot by testing localization algorithm using Kalman filter. Because of GPS position fixes are inaccurate and at times may not be available, other navigation aids are used in conjunction with GPS to enhance system performance. Dead-reckoning sensors can not be used alone for indefinitely long periods, since errors grow without bound. They accurately measure changes in a vehicle's position over short time (can be used alone when GPS fixes become unavailable for short periods). The nature of the errors in GPS position fixes is somewhat different than that of the errors in dead reckoning. The errors appear in GPS position fixes and the errors in dead reckoning are complementary in nature. Proper fusion of the GPS position fixes with the dead reckoning sensor data can take advantage of the complementary errors producing positioning performance better than either type of data alone [5]. In [6] and [7], the ATRV-Jr platform manufactured by IRobot used in the experiment is very expensive compared to the cost of our platform.

Experimental Results and Conclusion:

Kalman filters have been efficiently used for state estimation [9]. A common filter type, is the odometric filter where readings from the odometry system on the robot are used together with the geometry of the robot movement as a model of the robot. In order to be able to use the Kalman filter the process model in equation (1) needs to be discretized first and then linearized. The discretized process model is explained in [8]. The detailed description of Kalman filter setup is described in [12].

In this experiment we initiated the robot at $[x(0) \ y(0) \ \theta(0)] = [0 \ 0 \ 0]$ and set the target position at $(x_t, y_t) = (20, 9)$.

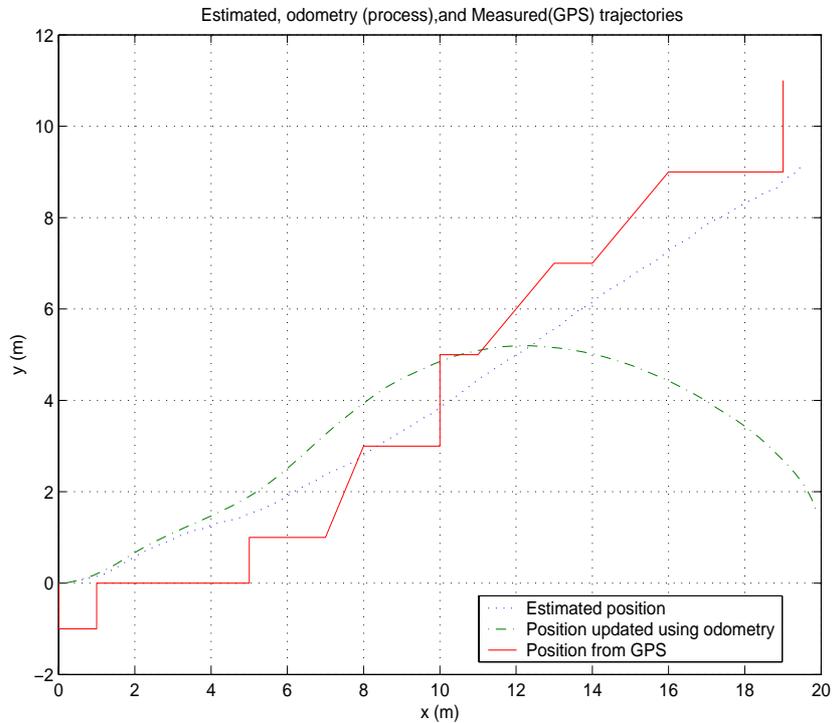


Figure (7): Estimated, measured and process trajectories.

The robot is commanded to reach target position using controller described in [11], which is based on input output feedback linearization [10]. The position was estimated using the Kalman filter. We do not have a ground truth for real trajectory traversed by the robot, however, the final position reached by the robot was within 1 meter error from the position of the target. Fig. 7 shows the trajectories computed from estimated state of Kalman filter, measured position from GPS, and odometric updated position. The odometric trajectory is updated using the kinematic model

An Educational Velocity Controlled Robot Vehicle

without taking the advantage of GPS and compass. The odometric state is correct in the beginning; however, in the middle of the path the error accumulated to give completely wrong y position at the end. The GPS readings are correct and within the device accuracy. The estimated position agrees with the observation that the robot follows a straight line toward the target. We have demonstrated that good results of localization are obtainable by using only an inexpensive well calibrated dead reckoning sensors and an inexpensive commercial GPS unit. This shows the effectiveness of the velocity controlled robot described in this paper.

ACKNOWLEDGMENTS:

I wish to express my sincere appreciation to the Fulbright Scholarship Program and AMIDEAST staff. Special thanks to my advisor Dr. R. Fierro at Oklahoma State University.

REFERENCE:

1. J. Laumond Robot Motion Planning and control. Springer, (1998).
2. R. Fierro and E. Edwan. The OSU multi-vehicle coordination testbed, in Proc. IEEE Midwest Symposium on Circuits and Systems (MWSCAS), Tulsa, OK, Aug. (2002).
3. P. McKerrow and D. Ratner. Calibrating a 4-wheel mobile robot, in IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, Sept./Oct. (2002), pp. 859 864.
4. N. S. Nise. Control Systems Engineering, 2nd ed. NY: Addison-Wesely, (1995).
5. E. Abbott and D. Powell. Land-vehicle navigation using GPS, in The Proc. of the IEEE, vol. 87, no. 1, Jan. (1999), pp. 145- 162.
6. S. Panzieri, F. Pascucci, and G. Ulivi. An outdoor navigation system using GPS and inertial platform, IEEE/ASME Trans. Mechatron., vol. 7, pp. 134 142, June (2002).
7. R. Thrapp, C. Westbrook, and D. Subramanian,. Robust localization algorithms for an autonomous campus tour guide, in Proc. IEEE International Conf. Robot. Automat., vol. 2, (2001).
8. C. Wang. Location estimation and uncertainty analysis for mobile robots, in Proc. IEEE International Conf. Robot. Automat., Apr. (1988), pp. 1231- 1235.
9. R. Brown and P. Y. C. Hwang. Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, 3rd ed. John Wiley Sons Inc., (1997).
10. A. Isidori. Nonlinear Control Systems, 3rd ed. London: Springer-Verlag, (1995).
11. J. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots, in Proc. IEEE International Conf. Robot. Automat., vol. 20, Leuven, Belgium, May (1998), pp. 2864- 2869.
12. E. Edwan and R. Fierro. A low cost modular autonomus robot vehicle, in Proc. 38th Southeastern Symposium on System Theory (SSST 2006) Cookeville, TN, USA, March 5-7, (2006), pp. 245- 249