Speed Control for a Mobile Robot

Kaylan Kolli, Sreeram Mallikarjun, Krishnamohan Kola and Ernest L. Hall

University of Cincinnati

Center for Robotics Research, ML # 72

Cincinnati, OH 45221

ABSTRACT

Automated Guided Vehicles (AGV's) have many potential applications in manufacturing, medicine, space and defense. The purpose of this paper is to describe exploratory research on the design of a speed control for a modular autonomous mobile robot controller. The speed control of the traction motor is essential for safe operation of a mobile robot. The challenges of autonomous operation of a vehicle require safe, runaway and collision free operation. A mobile robot test-bed has been constructed using a golf cart base. The computer controlled speed control has been implemented and works with guidance provided by vision system and obstacle avoidance using ultrasonic sensors systems. A 486 computer through a 3-axis motion controller s upervises the speed control. The traction motor is controlled via the computer by an EV-1 speed control. Testing of the system was done both in the lab and on an outside course with positive results. This design is a prototype and suggestions for improve ments are also given. The autonomous speed controller is applicable for any computer controlled electric drive mobile vehicle.

Key words: automated guided vehicle, speed control, safety, motion control.

1. INTRODUCTION

The design of a mobile system is a challenging task. The specific challenge of designing an intelligent controller is in determining what information is needed, how to measure it and how to use this information in a manner that will satisfy the performance specifications of the machine. The design specifications were to build a robot that follows a line, avoids obstacles, and adapts to variations in terrain. This implied the design of separate subsystems with discrete design objectives integrated in an upper level control logic that enables the robot to function as an integral system meeting all the performance requirements.

Jones and Flynn[1] give an excellent introduction to the implementation of small mobile robots. Muir and Neuman[2] give a rigorous mathematical treatise on the kinemaic and dynamic modeling of mobile robots. Cheng and Rajagopalan[3] also consider kinematic modeling for mobile robots. Nikam, et al[4], describes the overall design of the UC mobile robot.

At the subsystem level, the primary design considerations included the selection of equipment with the desired functional and operational features as well as reliability, commercial availability and affordability. Equally important was the compatibility of the software that controlled these units and their interfaces. Also all the subsystem components have been chosen to be modular in design and independent in terms of configuration so as to increase adaptability and flexibility. This, in fact is a unique feature of the design since it enables replacement of existing components with more sophisticated or suitable ones, as they become available. To ensure desired performance of the individual sub-systems, several unique approaches were tried. These include, the implementation of a Fuzzy Logic Controller for obstacle avoidance and a novel three-dimensional algorithm that outperforms the existing methods for line following. In the design and development phase of the different systems various analytical, experimental and computational methods were utilized.

The purpose of this paper is to describe the design of the speed controller for the traction motor of the mobile robot. The system design and speed control is described in Section 2. The experimental results are described in Section 3. Conclusions and recommendations are given in Section 4.

2. SPEED CONTROL

2.1 System overview

An autonomous mobile robot is a sophisticated, computer controlled, intelligent system. The adaptive capabilities of a mobile robot depend on the fundamental analytical and architectural designs of the sensor systems used. The mobile robot provides an excellent test bed for investigations into generic vision guided robot control since it is similar to an automobile and is a multi-input, multi-output system. The major components of the robot are: vision guidance system, steering control system, obstacle avoidance system, speed control, safety and braking system, power unit and the supervisor control PC.

Following is a brief description on the design and development of the main subsystems of the mobile robot. A block diagram of the system is shown in Figure 1.

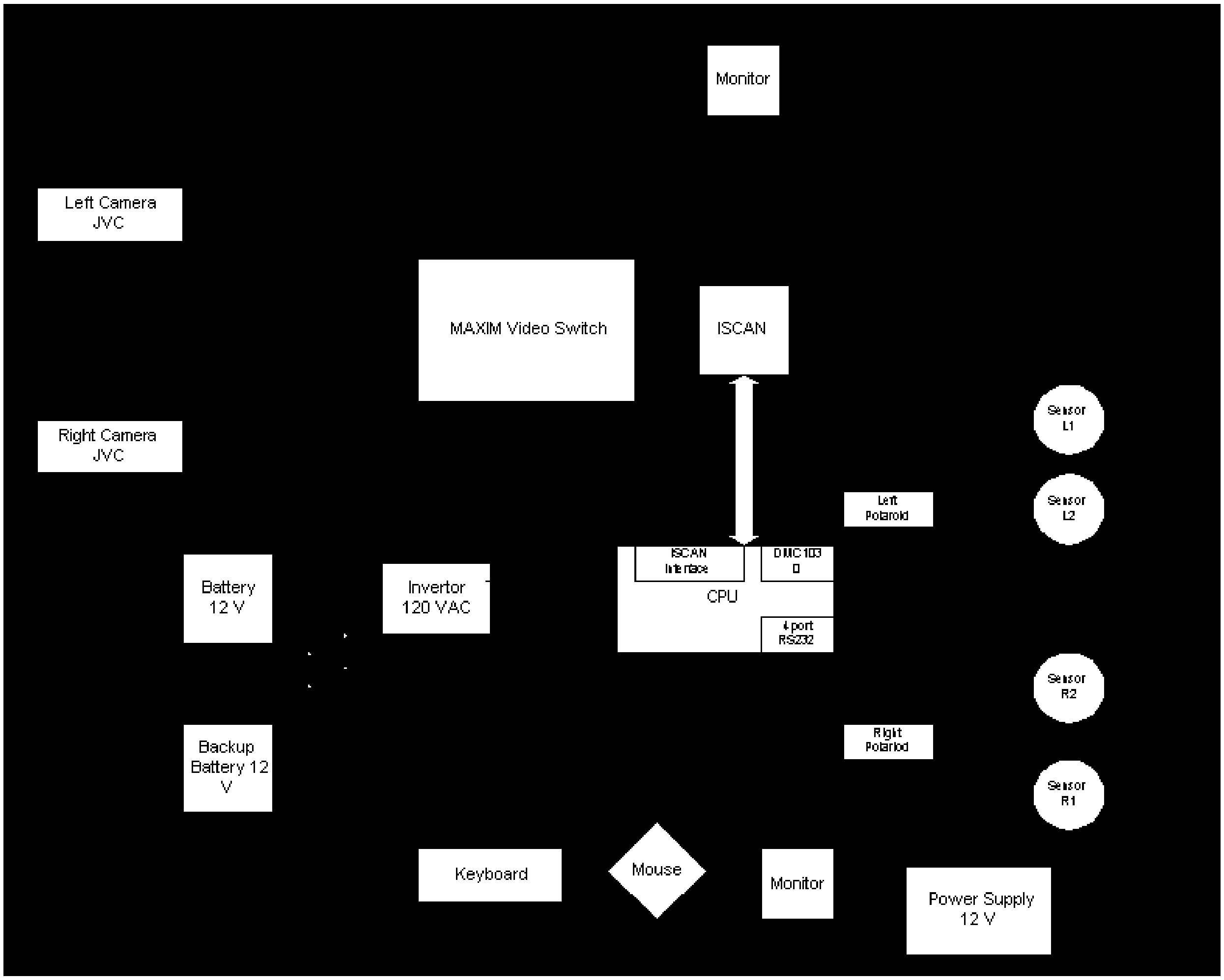

Figure 1. System block diagram

2.2 Speed control system

The robot base is an E-Z-Go[4] golf cart. A 36-volt, 55 amp traction motor drives this cart. Several designs were considered for controlling the large amount of power required for the traction motor, including: relays, power MOSFET's, and Insulated Gate Bi-polar Transistors (IGBTs). For the final design we choose the GE EV-1[5] speed controller. This is a commercial controller designed for forklift and other industrial electric vehicles. The EV-1 is a silicon con trolled rectifier (SCR), pulse width modulation (PWM) controller with a 0-10 volt control signal coming from the Galil [6] DMC-1030 motor controller and sufficient output to drive the traction motor at full power. To complete the control loop, a potentiometer is mounted above the front wheel. The position feedback signal is numerically differentiated to provide a velocity feedback signal.

Figure 2. Overall traction control system

Safety is of primary concern in the system design. For safety reasons, the EV-1 has a set of three sequential switches, which must be activated in order for it to run as shown in Figure 2. That is, a single switch cannot turn on the machine. A sequence of three switches must be activated in a prescribed order. Also in the main power loop, a solenoid is connected through three E-stops, the remote stop, as well as through the computer. This design should prevent any possible *runaway* of the vehicle since it provides a disconnect of power to all systems and application of the brake, not just breaking the low current control circuits.

The speed is controlled by the computer through the Galil motion control that outputs a signal that varies the voltage across the R5 to R4 connections on the EV-1. A 0 voltage across these terminals will cause the motor to go at full speed. A maximum voltage of -4.5 volts will cause the motor to go at creep speed. The emergency stop circuit includes a normally closed solenoid switch in series with this power circuit. If the e-stop switch is activated, the solenoid circuit opens and cuts power to the traction motor.

The speed control circuit is energized by closing three switches in the sequence -- Ready, Set and Go. These switches are connected to the GE EV-1 and for a forklift would be connected to the key switch, the seat switch and the accelerator/neutral switches. The EV-1 silicon controlled rectifier (SCR) operation controls the traction motor in the following manner. The control circuit supplies a gate pulse to a rectifier, turning it to the on state. This permits current to flow from the positive battery terminal through the SCR positive terminal (P), through the capacitor, through the rectifier, the IX coil to the SCR terminal T2 then either through the forward (F) or reverse (R) relay to the traction motor stator or field then through SCR terminal (A1) through the traction motor armature and through SCR terminal A2, the current sensor, to SCR terminal N and then back to the negative battery terminal.

The interface for the system was implemented using a Galil 1030-motion control computer interface board. A Galil breakout board permits the amplifier and encoded to be easily connected. The steering mechanism gets its input from the angle to be moved from the fuzzy logic controller. This controller takes the two inputs: the angle from the obstacle avoidance and the angle from the vision algorithm, and determines an output steering angle. Feedback is provided at a low level by a position encoder and at a high level by the vision and sonar systems.

2.3 Analytical design

The operations of the velocity loop where the motor velocity is sensed by a tachometer and is feedback to the comparator.

The transfer function between the input voltage, V, and the output velocity ω is of the form:

Where the constants are defined as:

Kt is the motor torque constant or ratio of the output torque to the input current,

Ka is the amplifier gain or ration of the output drive current to the input voltage.

Kg is the tachometer gain or ratio of output voltage to input speed,

J is the combined inertia of the motor and load.

The time constant τ_1 is given by:

The Galil motion controller permits one to implement a PID (Proportional, Integral, and Derivative) controller through an equivalent digital filter. The form of the digital filter is:

D(z) = K*(z-A)/z + C*z/(z-1)

The sampling time is T. The filter parameters K, A and C are selected by the instructions KP, KD, and KI and are related as follows:

K = KP+KD

A = KD/(KP+KD)

C = KI/8

C = KI/8

This digital filter includes a lead compensator and an integrator. It is equivalent to a continuous PID filter with transfer function G(s) where:

$G(s) = P + sD + I/s$

Where

$P = K(1-A) = KP$

$D = T*K*A = T*KD$

$I = C/T = KI/(8*T)$

The filter parameters may be determined either analytically or experimentally. One analytical method is described by Galil[9] and lets one select a phase margin at a given crossover frequency depending on the system parameters. In our case most of the system parameters such as the motor torque constant Kt, the motor and load inertia, J, the motor resistance< r, the amplifier constant, Ka, the encoder constant N were not known. Therefore an experimental design was performed using the SDK1 000 servo design software supplied by Galil.

2.4 Experimental design

The servo design kit program SDK1000[6] permits one to determine the values for the digital compensation filter through an experimental process in which system characteristics are measured online. That is, a stimulus signal is sent to the motor and load and the response is measured through its encoder. However, this program could not be used directly since there was no way to close the speed loop without the machine moving. Therefore, a trial and error process was used to determine gain values. The following values were found to perform satisfactorily.

KP=739

KD=27.5

KI=33

These values and others were set by a C++ program that is shown below.

```
void set_upDMC(){

//This function when called sends the initial settings for proper operation

//of the steering and drive motors.

submitDMC("KD,27.5"); Set derivative constant KD

submitDMC("KP,739"); Set position constant KP

submitDMC("KI,33"); Set integral constant KI

submitDMC("TL,9.98"); Set torque limit

submitDMC("FL,90"); Set forward position limit

submitDMC("BL,-90"); Set backward position limit

submitDMC("IT,1"); Set independent time constant for smooth acceleration

submitDMC("SP,20000"); Set slew speed to 20000 counts/second

submitDMC("AC,9216"); Set acceleration rate

submitDMC("DC,9216"); Set deceleration rate


}

#endif
```

The following program, SPEED.DCP, was used to communicate with the Galil controller using its TALK2BUS program.

David Perdue wrote the SPEED.DCP in 1996. To execute this program, one types TALK2BUS, then downloads SPEED.DCP, then executes the program by typing: XQ.

Statement Comment

```
#INIT Define label INIT

OUT=0 Initialize OUT

INC=0 Initialize INC

TWAIT=80 Initialize TWAIT to 80 msec
```

GAIN=.0005 Set GAIN to .0005

GAINACC=.000 Set GAINACC to 0

VEL=0 Set VEL to 0

VELTAR=0 Set VELTAR to 0

CB1 Clear bit number 1 on the output port. This bit is connected to a relay that controls the forward/reverse contactor.

DIR=1 Initialize DIR to 1 for forward direction

#SPEED Define label SPEED

P1=_TPZ Set P1 equal to the Tell Position value of the Z axis

T1=TIME Set T1 equal to TIME

WT TWAIT Wait for TWAIT milliseconds

P2=_TPZ Set P2 equal to the Tell Position value of the Z axis

VELOLD=VEL Set VELOLD to velocity value

VEL=(P2-P1)/TWAIT*1000 Compute new velocity value VEL from the difference quotient

INC=(VELTAR-VEL)*DIR Set increment to the difference in the velocity target and actual times direction

ACCEL=(VEL-VELOLD)/TWAIT*1000*DIR Compute acceleration from the difference quotient

JS#REV, VELTAR<0 Jump to reverse subroutine if velocity target, VELTAR, is negative

JS#FOR, VELTAR>0 Jump to forward subroutine if velocity target is positive

OUT=(INC*GAIN)+(ACCEL*GAINACC)+OUT Compute new OUT value

JS#MAX, OUT>9 Jump to subroutine MAX if out is greater than 9

JS#NEG, OUT<0 Jump to subroutine NEG if OUT is negative

OF OUT Offset command sets a bias in the motor command output or returns a previously

Set value. The value of OUT should be in the range of –128 to 127 decimal

JP#SPEED Jump to label SPEED

EN End of main program

#ZERO Label for subroutine ZERO

```
INC=0 Set INC equal to 0

EN End subroutine ZERO

#MAX Label for subroutine MAX

OUT=9 Set OUT equal to 9

EN End of subroutine MAX

#FOR Label for subroutine FOR

CB1 Clear bit 1 on the output port to go forward

DIR=1 Set DIR equal to 1 for forward motion

EN End of subroutine FOR

#REV Label for subroutine REV

SB1 Set bit 1 on the output port to go in reverse

DIR=-1 Set DIR to -1

EN End of subroutine REV

#LIMIT Label LIMIT

JS #ZEROP, OUT<. 1 Jump to subroutine ZEROP if OUT is less than 0.1

INC=.2*OUT Otherwise double the OUT value

EN End of subroutine LIMIT

#ZEROP Label for ZEROP

OUT=.1 Set OUT to lower limit

EN End of subroutine ZEROP

#NEG Label for subroutine NEG

OUT=0 Set OUT to 0

EN End of subroutine NEG
```

The END command is used to designate the end of a program or subroutine. If a subroutine has been called by the JS instruction, the EN command will return execution to the instruction after the JS command. If there is no subroutine being executed, the program terminates. During operation of the vehicle, the SPEED loop runs continuously. The values such as the target velocity, VELTAR, are changed by the main control program through either keyboard entry in manual mode or program control in automatic mode.

3. RESULTS of TESTING

After extensive laboratory testing of individual subsystems, an oval outdoor test track was constructed to simulate the contest track with double lines, 4 inches wide spaced 10 feet apart with dashed segments and obstacles. The first outside test was conducted on May 10, 1997. The performance of the speed control system was made in several different conditions including runs in the rain.

1. The speed control system operated as designed with variable speed in both forward and reverse, manual and automatic operation.
2. In some cases the machine would start at a very high speed, a lurch. This was due to the fact that when the computer power was off, a 0 voltage was presented to the R4-R5 terminals on the EV-1. With 0 volts, the EV-1 goes at full speed. Thus, if the traction power was turned on but the computer power was not turned on, the robot could lurch forward or in reverse depending on the state of the forward/reverse contactor.
3. Emergency stops were used on several occasions and the normal mode of operation required one person to be very near the e-stop.
4. No runaway situation was observed.
5. The slew speeds are controlled both by analog controls on the EV-1 and the digital control on the Galil controller board. Both sets of controls need to be carefully adjusted. Also, when an adjustment is needed, the machine was stopped, put on jacks and the analog pots tested first.
6. The mounting of the velocity feedback device on the front wheel made it difficult to test the speed loop with the robot on jacks. This also prevented an easy method of measuring the controlled speed. In future designs, the tachometers should be placed on the read drive wheels and a speedometer added.
7. Timing the robot's travel between 2 fixed points on a test track tested the speed. This method was adequate for determining that the robot was not moving faster than 5 mph.

4. CONCLUSIONS and RECOMMENDATIONS

A speed control system for a large traction motor has been designed, constructed and tested. The SCR device, the EV-1, performed very well and includes several safety features and other adjustments. The computer control device, the Galil DMC-1000 also worked extremely well. The servo design software permitted the tuning of the PID controller in an experimental manner.

The entire design team became alert to safety conditions and dangerous situations such as the robot lurching upon startup, the potential for runaway and collisions. Design considerations should be given to several situations in which the mobile machine could become dangerous. The condition of "runaway", in which the vehicle is totally out of control, should be prevented. This situation could occur from a short circuit in the control contactors. In this design, the emergency stops remove power from the traction motor to counter this situation. The condition of "lurching" on startup or shutdown was also discovered. In this design, this can be avoided by manually powering up the computer control before the traction motor and powering down the traction motor before the computer control. Adjustments of the speed control acceleration pots are also done with the robot jacked up off the ground. An automatic circuit should be designed for this operation. Collisions occurred during the testing due to the steering control. A safety bumper or other features such as obstacle avoidance sensor control over the traction motor to prevent collisions should be added in future designs.

Building an experimental testbed mobile robot is an excellent learning experience and brings out situations that cannot be easily anticipated from a theoretical analysis alone.

ACKNOWLEDGEMENTS

This work has been a continual learning experience and could not have been completed without the devoted work and contributions of the previous team members especially: David Perdue, Bradley Matthews, Tayib Samu, Mike Ruthe meyer, Alan Lewis, Malik Spencer, Sanjeev Gupta, Fred Reckelhoff, Todd Brehm, Randy Smith and Rob Hicks. The authors would like to thank the following staff personnel for their technical assistance; Bo Westheider (electronics), Dave Breheim (machinist), Perry Morgan (computer) and Larry Schartman (computer). The vehicle has been sponsored in part by several generous contributions and we thank our current and past sponsors: 1997, Dr. Ray Asfahl, SME, ICRC, UC; 1996, SME, UC, ICRC; 1995, Michel Tire Comp any; 1994, GE, Ohio Board of Regents; 1993, Textron, IBM, GM, NSF.

REFERENCES

[1] J.L. Jones and A.M. Flynn, Mobile Robots, A.K. Peters, Wellesley, MA, 1993, pp. 210 –224.

[2] P.F. Muir and C.P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," **Journal of Robotic Systems**, 4(2), 1987, pp. 281-340.

[3] R.M.H. Cheng and R. Rajagopalan, "Kinematics of Automated Guided Vehicles with an Inclined Steering Column and an Offset Distance: Criteria for Existence of Inverse Kinematic Solution," **Journal of Robotic Systems**, 9(8), 105 9-1081, Dec. 1992.

[4] Umesh Nikam, Nikhil Kelkar, Kaylan Kolli, Krishnamohan Kola, Wen-Chuan Chiang, Raymond Ande, Sreeram Mallikarjun and Ernest L. Hall, "Design of a Mobile Robot for the 1997 AUVS Competition," University of Cincinnati Techni cal Report, June 1997.

[5] General Electric, EV-1 SCR Control Manual, Charlottesville, Virginia(1986).

[6] Galil Inc, DMC-1000 Technical Reference Guide Ver. 1.1, Sunnyvale, California(1993).