Dynamics of the Titan four-wheel drive mobile robot with floating Ackerman steering

Danny Ratner
Intelligent Robotics Laboratory
School of Information Technology and Computer Science
University of Wollongong
NSW 2522
Australia
email: dr19@uow.edu.au
home page: http://www.uow.edu.au/~dr19/

Phillip M^{c}Kerrow
Intelligent Robotics Laboratory
School of Information Technology and Computer Science
University of Wollongong
NSW 2522
Australia
email: phillip@uow.edu.au
http://www.itacs.uow.edu.au/people/phillip/robotics.html

Abstract

An Australian designed a new steering mechanism for a 4-wheel drive chair. This mechanism achieves differential velocity steering without skid steering. In this paper, we analyze the kinematics and dynamics of this design to develop a model of the robot. We simulated this dynamic model with Simulink and conducted experiments to validate it with a robot controlled by LabVIEW.

1 Introduction

Dr. Bernard Silderstein is a paraplegic who wanted a wheel chair that he could drive on sports fields, beaches etc. Mr. Jeffery Pagett designed and patented [Pagett, 1996] a new steering mechanism for a 4-wheel drive (4wd) wheel chair. This mechanism achieves differential velocity steering without skid steering. We have modified a commercial 4wd wheel chair (Titan) – into a 4wd outdoor mobile robot, Figure 1.

Figure 1. Titan 4 wheel-drive wheel-chair

Our research goal is to navigate this robot using landmarks recognition. The landmarks are plants and the edges of paths. They are detected using a CTFM ultrasonic sensor array [Ratner & MacKerrow,1977]. To achieve the goal of autonomous navigation, we have implemented a controller with LabVIEW running on a Macintosh PowerBook. To measure the location of the robot, we have installed position and steering encoders and a gyro stabilized compass.

In this paper, we model the kinematics and dynamics of the robot. We simulate the robot with a model of the dynamics in Simulink. To validate this model, we conducted experiments with the robot. We plan to use this model as a simulation test bench for the development of the control and recognition algorithms in the Matlab/Simulink environment. After the algorithms are tested in design iterations they are implemented in the LabVIEW environment for real time experiments [Ratner & MacKerrow, 1998].

2 Mobile robot design

Titan is driven by 4 wheels with pneumatic tires. Each wheel is mounted on the shaft of a gear motor powered by a pancake motor (Figure 2). The motors are emf controlled in pairs (one pair per side) by PID controllers. The user manually controls forward velocity and steering direction with a 2D joystick.

Titan is steered by controlling the differential velocity between the two pairs (one on each side) of wheels. Thus, the joystick steering input is added to the velocity on one side and subtracted from the velocity on the other. Normally, differential velocity steering of a 4 wheeled vehicle results in skid steering. This is not desirable for a wheel chair, because skid steering can damage carpets and lawn, and scuff cork floors and concrete.

An innovation in steering design, that has been patented [Pagett, 1996], enables differential velocity to achieve Ackerman steering. Ideally, an Ackerman steering mechanism turns the front steering wheels by different angles so that the axes of all 4 wheels pass through a common point. As a result, no sideways force is applied to the wheels and hence, no skidding. In practice, car steering mechanisms only achieve this ideal at one steering angle.

Like a car steering mechanism, the mechanism used in Titan only achieves Ackerman steering at one steering angle. The steering mechanism consists of 4-bar linkage. This design achieves precise Ackerman steering at one

steering angle and generates some deviation at other steering angles. This mechanism was modeled geometrically using AutoCAD (Figure 3).

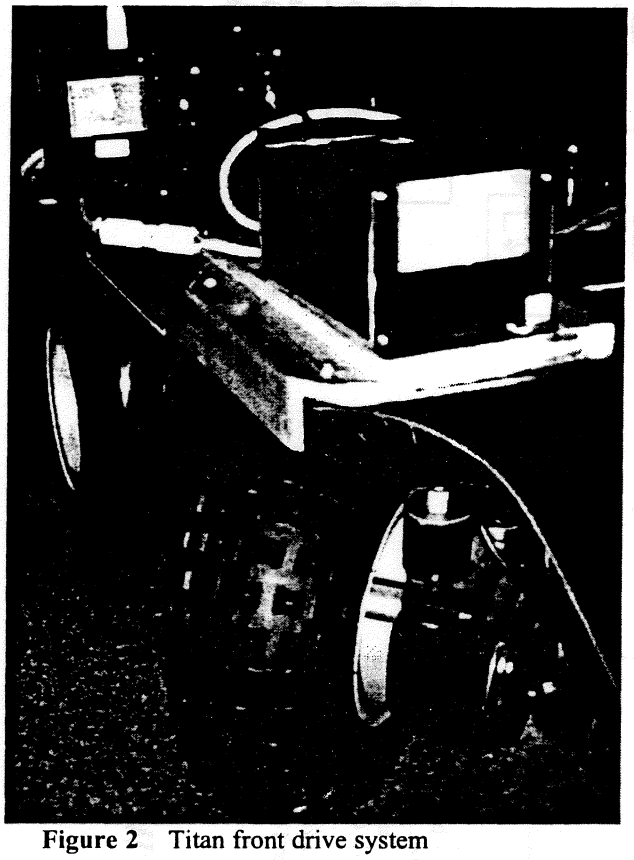


Figure 2 Titan front drive system

The gear motor for each front wheel is mounted on one of the 4 links (the two short links) as shown in Figure 4. These links are connected to the chassis by rotary joints, so the chassis forms the third (front) link. The fourth (back) link is an adjustable length rod that connects the motor links using rotary joints. The adjustment allows manual tuning of the angle of the front drive wheels so that they are parallel at zero degrees steering angle.

When the fourth link is open, both wheels are free to turn about the rotary joints connecting them to the chassis. This freedom is constrained by the forth link so that when one wheel turns the other also turns. When differential velocity is applied to the two pairs (one per side) of wheels, the front wheel with the lower velocity turns outward. The 4-bar linkage causes the front wheel on the outer side to turn with it, by a proportional amount. As a result, the configuration of the wheels approximates Ackerman steering. The robot turns in the direction of the lower velocity wheels with a turning circle whose radius is proportional to the differential velocity.

An additional feature of the design is that the axes of the wheels do not intersect the axes of the rotary joints that connect them to the chassis. In fact, the wheel axes are behind the joint axes (Figure 4). The result is that, when the robot is moving in the forward direction, the front wheels castor. We believe that this design achieves a forward stabilizing effect, making steering in the forward direction more positive. Unfortunately, the castor effect destabilizes steering in the reverse direction. Consequently, the joystick is physically constrained to only allow small steering angles in use.

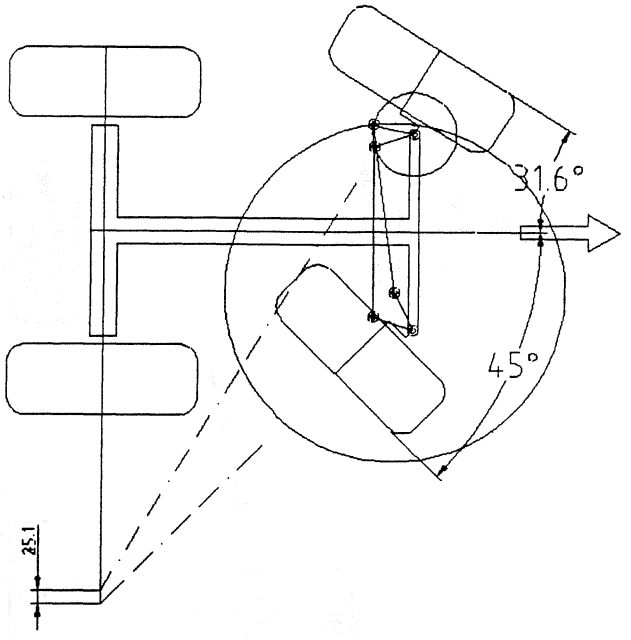


Figure 3 Deviation from ideal Ackerman steering

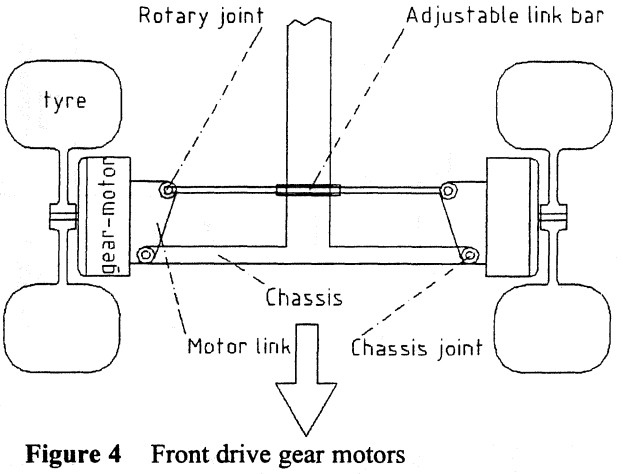


Figure 4 Front drive gear motors

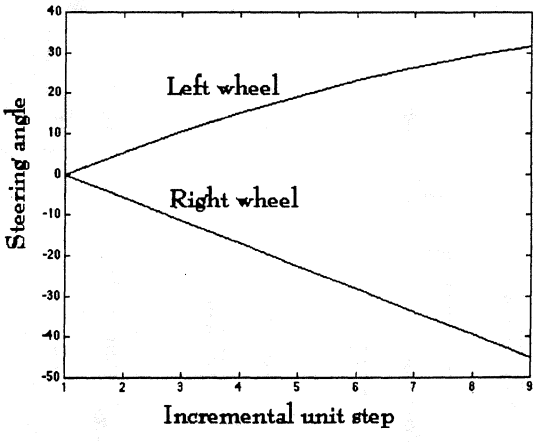
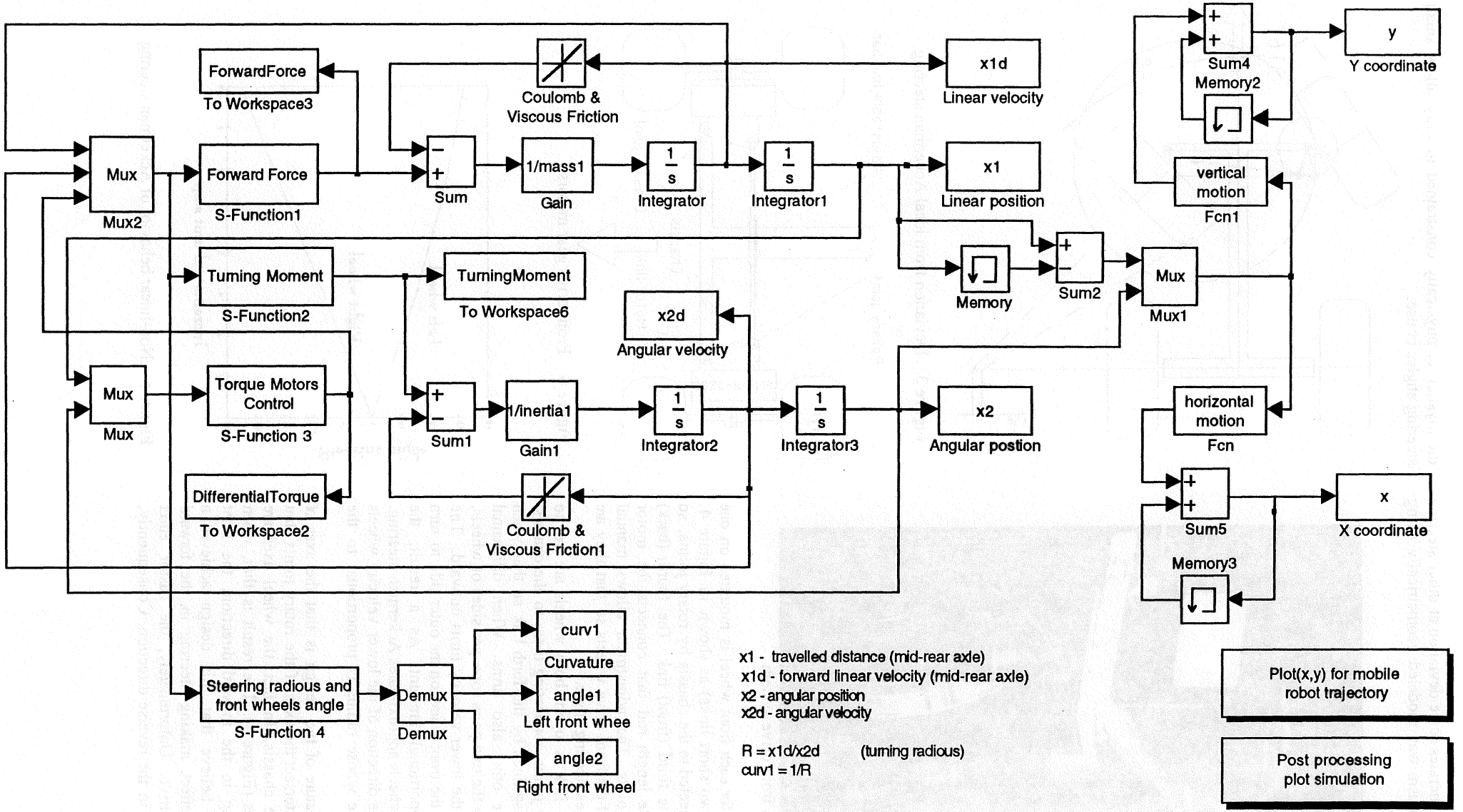


Figure 5 Non-linear behavior of Ackerman steering



Simulink dynamic model for Titan 4WD mobile robot

Figure 7

The non-linear transfer function relating the angles of the sides (on which the wheels are mounted) of a trapezoid were explored using a Matlab program. The 4-bar linkage was divided into two adjacent triangles which were solved numerically. This Matlab function was also integrated into the dynamic model. Figure 5 shows the angle of the left wheel (upper graph) as a function of the angle of the right wheel. (lower graph).

3 Kinematics model

The kinematics equations of a 4 wheel robot with floating front drive Ackerman steering (Figure 6) are similar to those of a turtle robot [McKerrow, 1999]. In a turtle robot, the robot frame is located at the center of the robot. The motion of a turtle robot can be described with three modes: pure rotation, pure translation and a combination of rotation and translation. In the case of Titan, the robot frame is located mid way between the rear wheels. Because of the angular limitation of the steering joints, the nonholonomic robot has only two modes of motion: pure translation, and combined translation and rotation [Sekhavat and others, 1998].

The angular velocity of the robot (Equation1) is found by dividing the differential linear velocity of the rear wheels by the rear wheel span. The turning radius (Equation 2) is found by dividing the linear velocity of the robot by the robot angular velocity.

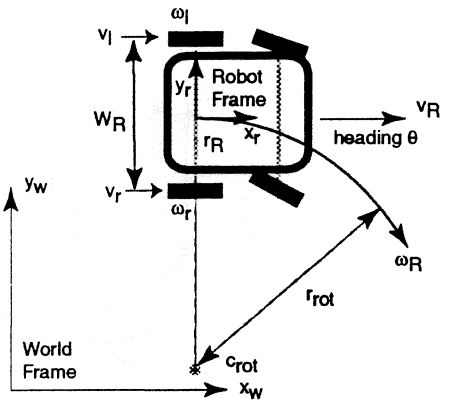


Figure 6 Kinematics model

$$\omega_R = (V_l - V_r)/W_R \qquad 1$$

$$r_{rot} = 0.5 \bullet (V_l + V_r)/\omega_R \qquad 2$$

4 Simulink model of Titan's dynamics

The dynamics of Titan is best explained by referee to the a Simulink model shown as a block diagram in Figure 7. The dynamic model in Figure 6 is based on the Newton -Euler formulation of dynamics. Simulink is a graphical programming language for control system modeling using block diagrams. To enable the modeler to include complex world models Simulink allows the user to program an algorithm in Matlab and define it as a Simulink S-Function block, which can be used in the block diagram model.

The model in Figure 7 includes two parallel loops: one for calculating linear motion (top) and one for calculating angular motion. This decomposition matches the inputs to the robot's controller. These velocities are calculated from the forces applied to the wheels.

The linear motion loop (top) calculates the linear position and velocity of the robot for the forces applied to the robot, less the resistive force due to friction and damping. The input to the loop is an applied force calculated by S-Function 1 from the torques applied to the wheels. These wheel torques are calculated by S-Function 3 to achieve the controlled trajectory.

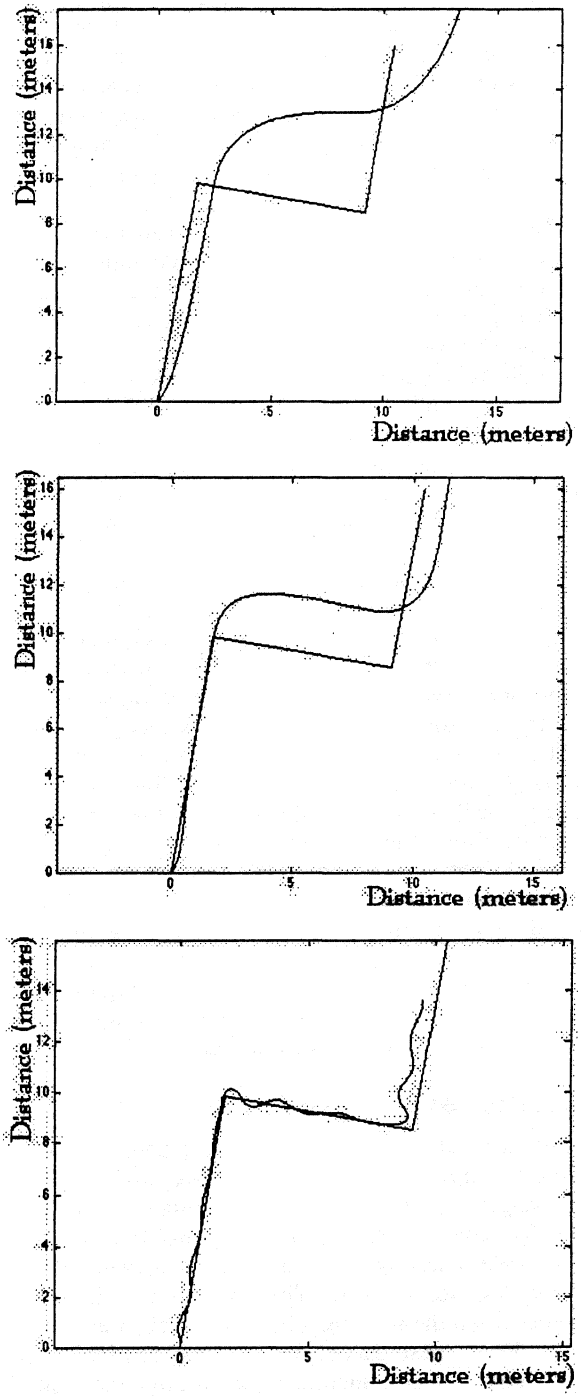


Figure 8 Simulated trajectory following with increasing loop gains (top to bottom), showing change in damping.

The angular motion loop is similar to the linear motion loop. Its input is a turning moment calculated by S-Function 2 from the torque applied to the wheels. On the right hand side of Figure 7, the x,y position of the robot is calculated from the motion values. At the bottom of the diagram, the turning radius and wheel angles are calculated for display.

The forces and torque applied to the robot are calculated in the S-Function blocks. The Torque Motors Control (S-Function 3) block reads the trajectory from a look up table. The table contains a distance and a bearing for each leg of the trajectory. It calculates the current location of the robot from the outputs of the above loops. From the trajectory and current location it calculates the error in reading. The output of this block is a differential torque proportional to heading error which is clamped to model the limitations of the motor control electronics of the robot.

The forward velocity (S-Function 1) block calculates the forces applied to the robot from the differential torque and the angle of the steering wheels. The angle of the steering wheels is calculated from the differential velocity between the two sides of the robot.

The Turning Moment (S-Function 2) block calculates the turning moment applied to the robot from the differential torque and steering angle. This system implements a Newton-Euler formulation because it calculates the acceleration (angular and linear) in each loop from the balance of the forces on the robot. The mass of the robot is approximately 100 kg.

Simplifying assumptions are used in the model. First, we assumed that the friction forces are large enough to avoid lateral sliding of the wheels. Second, we assume that the friction force resisting the turning of the wheels (and hence producing forward motion without skidding) is adequately modeled by the Simulink Coulomb and Viscous Friction block. Thirdly, we assume that the friction in the joints of the steering is negligible and the inertia of the links in the 4-bar linkage is small compared to the inertia of the robot.

The output from the simulation for a trajectory consisting of a right turn (from 100° heading to 190°) followed by left turn (from 190 degrees heading to 100 degrees) is shown in Figure 8. As the loop gain is increased, the response moves from overdamped to underdamped. The following three sub-figures (Figure 8) describe few behavior patterns as part of the sensitivity tests for parameters like the proportional control gain.

5 Steering angle coupling

The simulations shows a coupling effect between the turning moment contributed by the front drive wheels and the steering angle. In Figure 9, the turning moment produced by the front wheels is plotted against time for a 90° turn. The upper curve is the turning moment when the steering wheels are free to turn. The lower curve is the turning moment when the steering wheels are locked at zero steering angle. From these curves, we see that turning the front wheels during a turn increases the turning moment produced by the front wheels even though it reduces the motor torque.

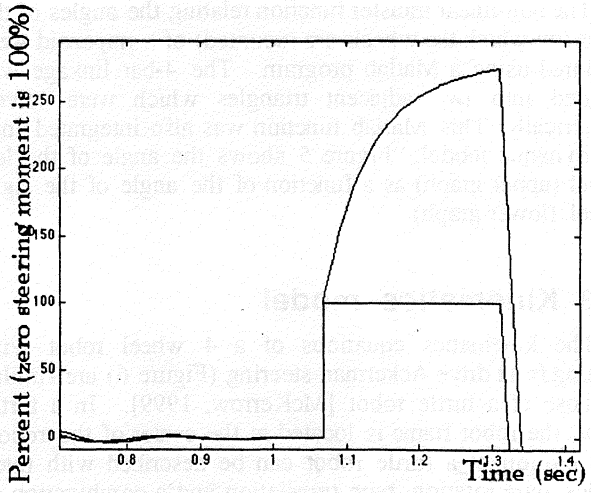


Figure 9 Effect of Ackerman steering on the front drive turning moment. The upper curve is the turning moment delivered from the front drive. The lower curve is for zero steering . Both are from clamped controller.

The effect of increased moment from reduced torque occurs because the effective moment arm of the outer wheel increases (from 185 mm to 316 mm) and the effective moment arm of the inner wheel decreases (from 185 mm to 27 mm) in Figure 10.

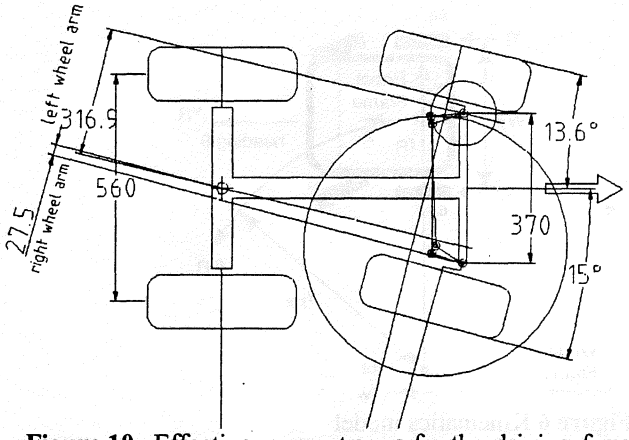


Figure 10 Effective moment arms for the driving forces of the front wheels due to Ackerman steering

7 Robot experiments

Titan's sensors and actuators are connected to a Macintosh PowerBook with National Instruments PCMCIA data acquisition card. The sensing and control software is written in LabVIEW [Ratner and M^{c}Kerrow, 1997]. A virtual control panel (Figure 11) was built to control this experiment. The location and heading of the robot were measured using encoders attached to the rear wheels and a gyro stabilized compass. The steering angle of the left front wheel was measured with an encoder.

In order to compare the motion of the robot with that predicted by the simulation, we developed a feedback loop to control vehicle heading with a LabVIEW formula node. The robot was driven along the same trajectory as used in the simulation. A trajectory is specified as a sequence of tuples. Each tuple specifies a heading and the distance to be

traveled in that direction. The trajectory in Figure 12 is 10m at 100°, 7.5m at 190°, and 7.5m at 100°. The linear velocity of the robot is fixed for the length of the trajectory, and the differential velocity is controlled with the feedback loop.

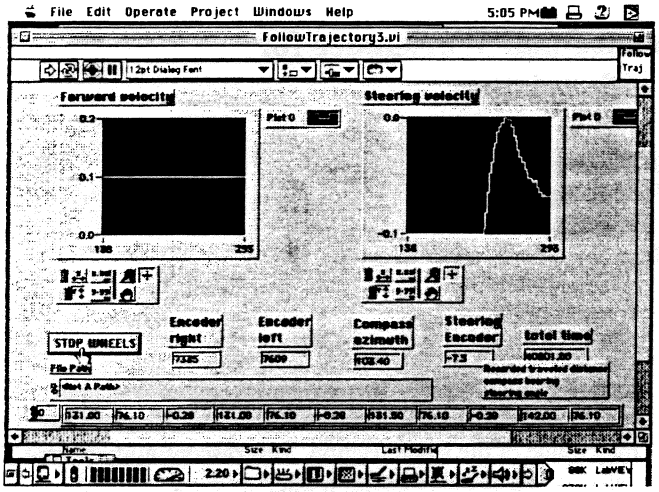


Figure 11 LabVIEW virtual control panel for the experimental set up

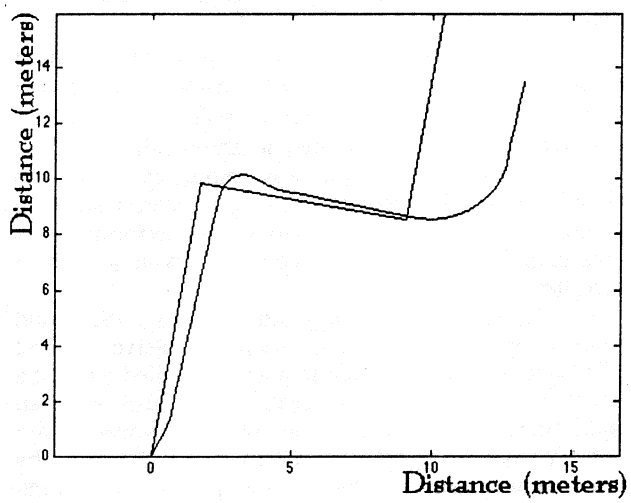


Figure 12 Trajectory of an outdoor experimental run

The feedback loop calculates the difference between the commanded heading and the heading measured with the compass. The formula node also clamps the input to the steering controller in order to limit the acceleration and hence, the motor current. Thus, it is equivalent to the calculations of turning moment in S-Function 2 of the simulation.

We studied the dynamic behavior of Titan by varying the loop gain and linear velocity. We found that the experimental response was similar to that of the simulation. Both transferred from overdamped to underdamped response on bearing changes as the loop gain was increased.

However, the actual trajectory in Figure 12 is different to the commanded trajectory. First, the robot must follow a curved trajectory to turn from one bearing to another. This results in a tracking error. Second, the experiment shows that the robot is slowly creeping to the right on the straight-line segments. This is due to a mismatch in the zero volt line between instrumentation on the robot, which we are currently investigating.

Thirdly, the right turn is much sharper and less damped than the left turn. This may be due to the mismatch mentioned above. However, we have not confirmed this.

8 Conclusion

The experiments show that the Simulink model is a reasonable representation of the robot dynamics. To model this robot we have used three complementary tools: AutoCAD for geometric modeling of kinematics, Simulink for block-diagram modeling of the dynamics, and LabVIEW for controlling the robot. This combination has enabled us to be very productive.

References

Pagett, J. M., 1966. "Steerable four wheel drive vehicle', U.S. Patent 5482125, Jan. 9, 1996.

Ratner D. and McKerrow, P.J. 1997. 'Landmark recognition with ultrasonic sensing for autonomous mobile robot navigation', FSR'97, December, Canberra, pp 552-558.

Ratner D. and McKerrow, P.J. 1998. 'Extracting Features from CTFM Sonar', in Chugani, M.L., Samat, A.R. and Cerna, M. eds, LabVIEW Signal Processing, Prentice Hall, ISBN 0-13-972449-4, pp 124- 131

McKerrow, P.J. 1999. 'Calibrating Mobile Robot Odometry with Ultrasonic Sensors', submitted to IEEE Transactions on Robotics and Automation.

[Sekhavat, Svestka, Laumond and Overmars] Multilevel Path Planning for Nonholonomic Robots Using Semiholonomic Subsystems. The International Journal of Robotics Research, Volume 17 Number 8 August 1998, pages 840-857.