



3 4456 0354318 2

ORNL/TM-11784  
(CESAR-91/04)

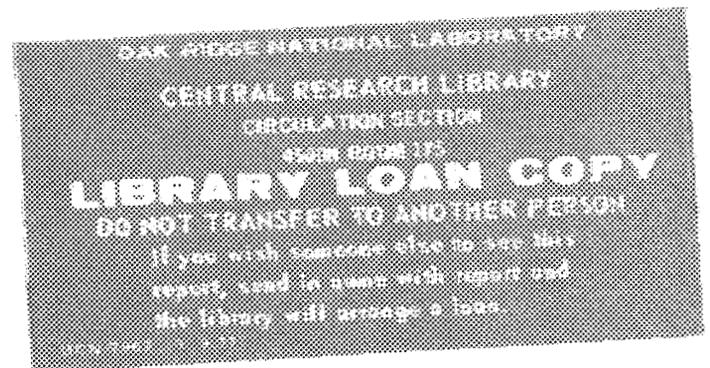
# ornl

OAK RIDGE  
NATIONAL  
LABORATORY

MARTIN MARIETTA

## A New Wheel Control System for the Omnidirectional Hermies-III Robot

D. B. Reister



MANAGED BY  
MARTIN MARIETTA ENERGY SYSTEMS, INC.  
FOR THE UNITED STATES  
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831, prices available from (615) 576-8401, FTS 826-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11784  
CESAR-91/04

Engineering Physics and Mathematics Division

**A NEW WHEEL CONTROL SYSTEM FOR THE  
OMNIDIRECTIONAL HERMIES-III ROBOT**

D. B. Reister

DATE PUBLISHED — April 1991

Research sponsored by the  
Office of Engineering Research Program  
Basic Energy Sciences  
U.S. Department of Energy

Prepared by the  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, Tennessee 37831  
managed by  
MARTIN MARIETTA ENERGY SYSTEMS, INC.  
for the  
U.S. DEPARTMENT OF ENERGY  
under contract DE-AC05-84OR21400



3 4456 0354318 2



## CONTENTS

ABSTRACT . . . . .	v
1. INTRODUCTION . . . . .	1
2. ARCHITECTURE . . . . .	5
2.1 PLATFORM GOAL . . . . .	7
2.2 WHEEL TARGET AND CONTROL . . . . .	9
2.3 WHEEL STATE . . . . .	9
3. WHEEL DRIVER AND RECKONER . . . . .	11
4. WHEEL CONTROLLER . . . . .	15
5. EXPERIMENTAL RESULTS . . . . .	17
6. CONCLUSIONS . . . . .	25
ACKNOWLEDGMENTS . . . . .	27
REFERENCES . . . . .	29
APPENDIX A . . . . .	31
APPENDIX B . . . . .	33

## LIST OF FIGURES

<u>Fig.</u>		<u>Page</u>
1	The omnidirectional HERMIES-III robot . . . . .	1
2	The context of the wheel control system . . . . .	2
3	The architecture of the wheel control system . . . . .	4
4	The world coordinate system for the platform . . . . .	6
5	The local coordinate system for the platform . . . . .	7
6	Path of the robot during the first experiment. The units of x and y are meters . . . . .	17
7	The velocity of the left wheel during the first experiment ( $v_L = v_L$ ). The units of $v_L$ are meters/second and the units of time are seconds . . . . .	18
8	The velocity of the right wheel during the first experiment ( $v_R = v_R$ ). The units of $v_R$ are meters/second and the units of time are seconds . . . . .	19
9	The target velocity (upper) and realized velocity (lower) for the left wheel . . . . .	20
10	The target (upper) and realized velocity (lower) for the right wheel . . . . .	21
11	Path of the robot during the second experiment. The units of x and y are meters . . . . .	22
12	Direction of the right wheel during the second experiment. The units of $\theta_R$ are radians and the units of time are seconds . . . . .	23
13	The velocity of the right wheel during the second experiment ( $v_R = v_R$ ). The units of $v_R$ are meters/second and the units of time are seconds . . . . .	24

## ABSTRACT

We have designed, built, and tested a new wheel control system for the HERMIES-III robot. HERMIES-III is a large mobile robot with omnidirectional steering that is designed for human scale experiments. During each cycle (at 20 Hz), the wheel control system moves the robot toward a goal and calculates the current position of the robot. The system has seven modes for moving to a goal and the goal may be changed during motion of the robot.



# 1. INTRODUCTION

A mobile robot's mission can be described by a sequence of postures (a posture is a position with an orientation). At each posture, the robot is at rest and may perform a task. In moving from one posture to the next, the sensors on the robot may detect unexpected obstacles. If unexpected obstacles block its path, the mobile robot must find a new clear path to its next objective. This paper describes the wheel control system for a mobile robot that could encounter unexpected obstacles and be required to quickly replan its path.

The motivation to consider this problem was provided by the need to plan paths for the third generation Hostile Environment Robotic Machine Intelligence Experiment Series (HERMIES-III) robot. HERMIES-III is a large robot designed for human scale experiments<sup>1</sup> (see Fig. 1). The chassis (1.6m  $\times$  1.3m  $\times$  1.9m) has two steering wheels and four corner caster wheels. Currently, the robot is tethered and weighs 820 kg (when battery powered, the weight will be 1230 kg). The manipulation system consists of the CESAR manipulator (CESARm) a 7-DOF compliant arm with all revolute joints, a spherical wrist, and a low friction back driveable drive train. The CESARm can reach 1.4m, has a load capacity of approximately 14 kg, and has an unloaded tip speed of 3.0 m/s. The kinematics and control of the robot are described in Jansen and Kress.<sup>2</sup>

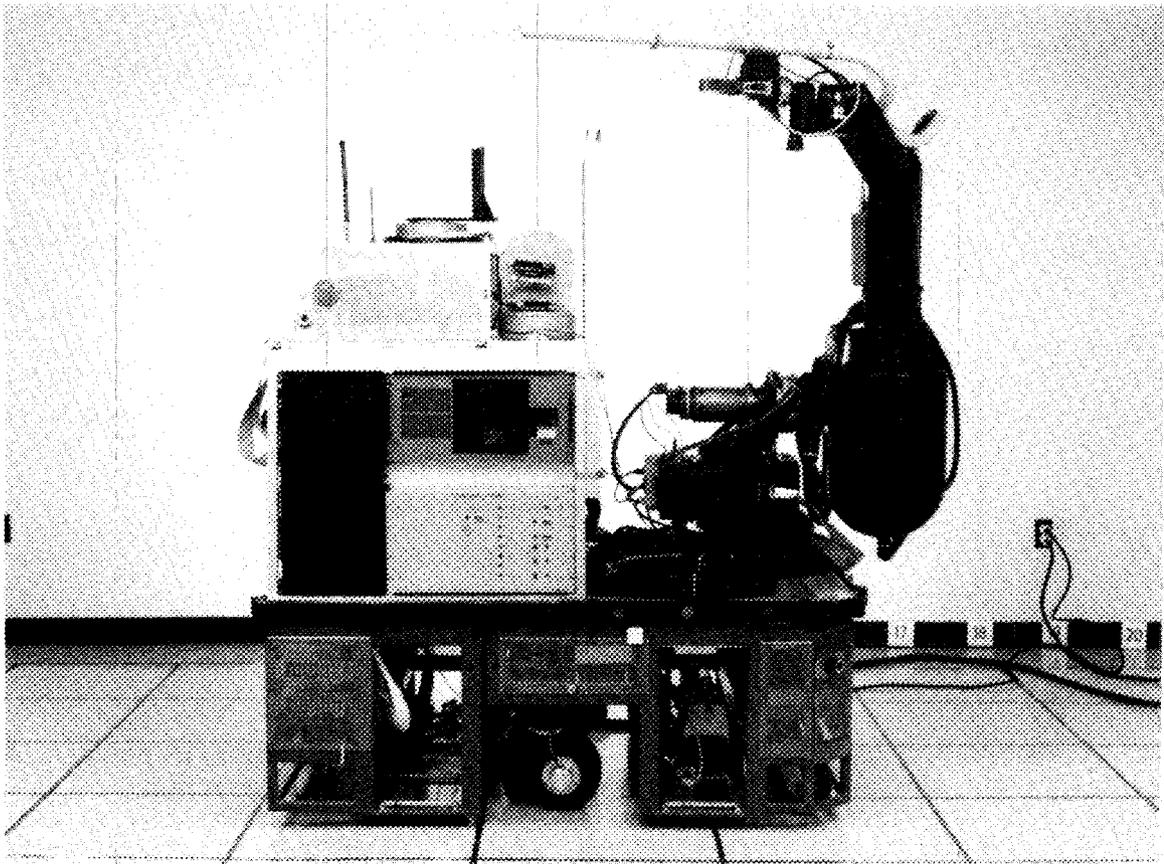


Fig. 1. The omnidirectional HERMIES-III robot.



sensors monitor the environment. If unexpected obstacles are detected, the path planner modifies the robots path to avoid the obstacles. The path monitor follows the revised path produced by the path planner and passes goals to the wheel control system. The wheel control system moves the robot and calculates its posture, which is passed to the other components of the motion system.

The architecture of the wheel control system is displayed in Fig. 3. The system has three components (controller, driver, and reckoner). Its higher level interface is with the path monitor and its lower level interface is with the motor driver. Given a platform goal (from the path monitor) and the wheel state (from the wheel reckoner), the wheel controller sends the platform posture to the path monitor and sends targets (for the wheel velocities and directions) and controls (wheel accelerations and steering velocities) to the wheel driver. The wheel driver sends setpoints to the motor drivers and reads the current encoder values from the motor drivers. Given the encoder values, the wheel reckoner calculates the current position and velocity of the robot. Communication between the five processes in Fig. 3 is through shared memory.

Our focus is on the wheel control system. The next section of this paper will describe the architecture of the system. The third section will describe the driver and reckoner, while the fourth section will discuss the controller. The fifth section will present some experimental results. The final section will review some of the lessons that were learned during the development of the system. The mathematical details of the kinematics of the robot will be developed in the two appendixes.

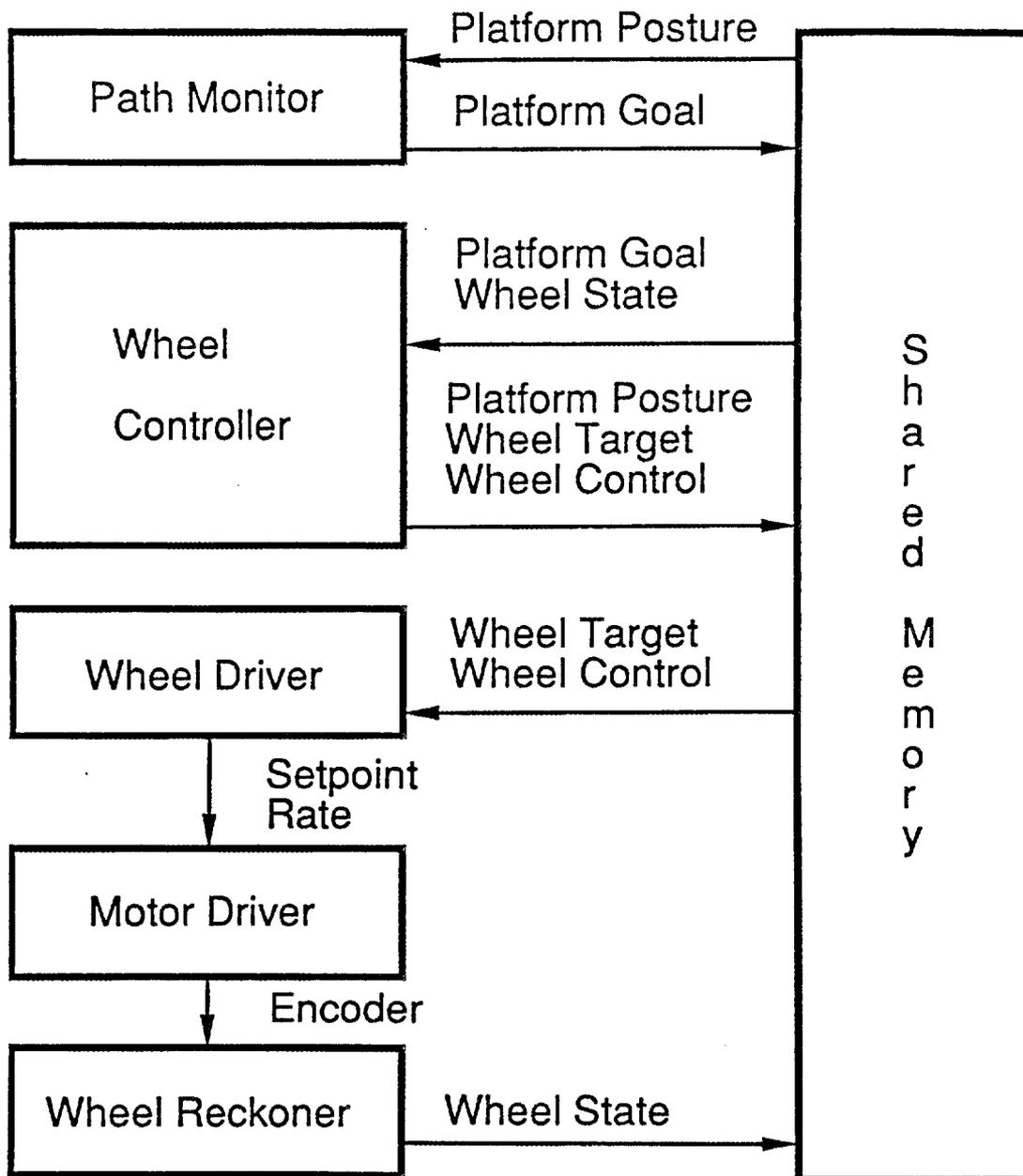


Fig. 3. The architecture of the wheel control system.

## 2. ARCHITECTURE

Figure 2 defines the context for the wheel control system and Fig. 3 provides names for the data structures that are used to pass information between the three components of the system. In this section, we will discuss what information should be contained in the three data structures. The most important data structures are the links to the higher levels (platform goal and platform posture). All of the other data structures in Fig. 3 are internal to the wheel control system. The information in the platform posture is the posture (position and orientation) of the robot. Specifying a platform goal is the only way that the higher levels of the motion system can move the robot. To identify the information that should be included in the platform goal, we begin this section by reviewing the types of motions that can be performed by an omnidirectional robot. By the end of this section, we shall identify seven types of motion.

A robot with two steerable drive wheels can perform several types of motions. When both wheels are pointing forward (perpendicular to the line joining the two wheels), the velocities of the two wheels are independent (both the direction of rotation and the speed of the wheels can differ). When either or both of the wheels are not pointing forward, the velocities of two wheels are dependent.

When both wheels are pointing forward and have the same velocity, the robot will move straight forward. If the wheels have opposite velocity, the robot will rotate. If both wheels move forward and the right wheel is faster than the left wheel, the robot will move in a circle and the center of the circle will be on the line joining the two wheels and to the left of the robot. If the wheels move in opposite directions, and the right wheel is faster than the left, the robot will move in a circle and the center of the circle will be between the two wheels and closer to the left wheel than to the right wheel. If we consider forward motion to be circular motion with the center of rotation at infinity, all motions by a robot with both wheels pointing forward are circular with the center of the circle on the line joining the two wheels.

Since the platform of a mobile robot is a rigid body, the distance between the two wheels is fixed. When either or both of the wheels are not pointing forward, the two wheels will be trying to pull the robot apart unless the velocities of two wheels are dependent. If both wheels point in the same direction, the velocities of the two wheels must be the same (unless the wheels point forward). If the two wheels point in different directions, the two lines that are the axes of rotation for the wheels intersect at the center of rotation for the robot. The center of rotation can be any point in the plane. In Figs. 4 and 5, the instantaneous center of rotation (ICR) is labeled P.

As the wheel velocities make the transition from independence to dependence, discontinuous changes in velocity may occur. For example, if the right wheel is steered and the left wheel is pointing forward, the ICR becomes the right wheel and the velocity of the right wheel must discontinuously go to zero. In general, smooth changes in the ICR will result in smooth changes in velocity. The steering angles for the two drive wheels on the HERMIES-III robot ( $\theta_R$  and  $\theta_L$ ) are limited to the range from minus 90 degrees to plus 90 degrees in the platform coordinate system (see Fig. 5). Consequently, some smooth trajectories for the ICR are prohibited.

For both a manipulator and a platform, it is useful to distinguish the joint variables from the Cartesian variables. For the platform, the Cartesian variables are position and orientation, while the joint variables are the rotation of each wheel about its axle and the steering angle of the wheel in the platform coordinate system.

## 6 ARCHITECTURE

The data structures that are the interface to the wheel control system (platform goal and platform posture) use Cartesian variables while the internal data sets (target, control, and state) use joint variables.

ORNL-DWG 91-6253

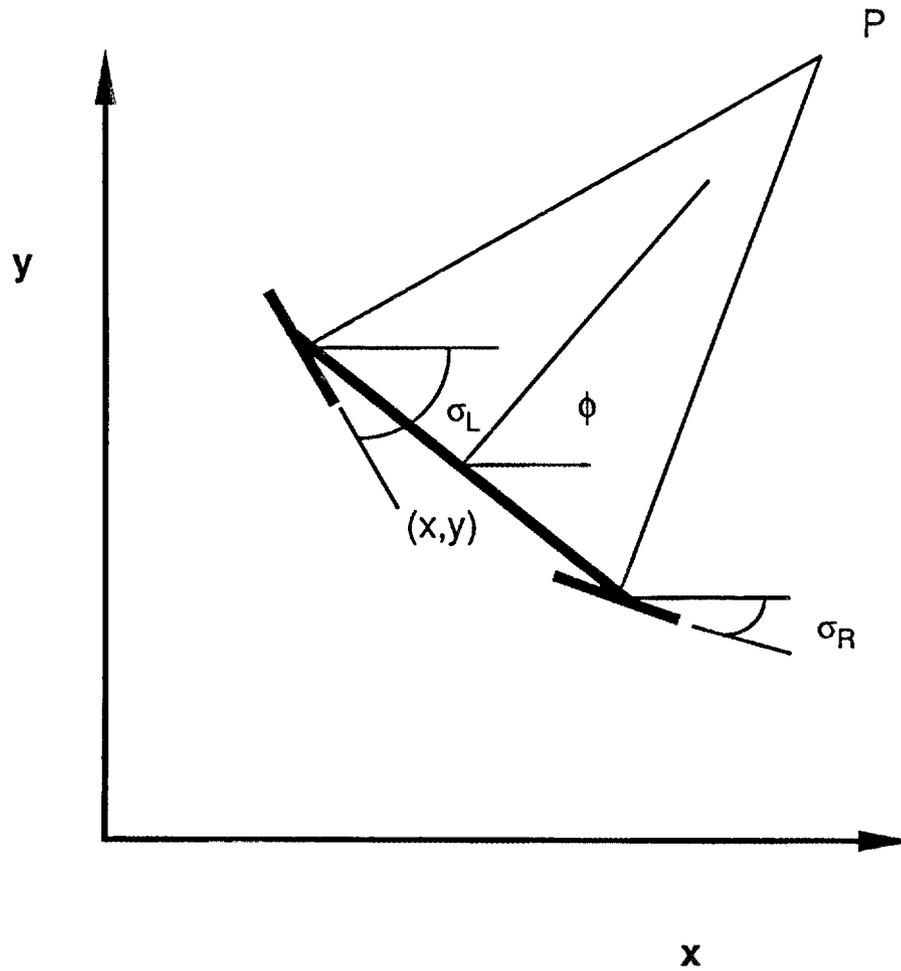


Fig. 4. The world coordinate system for the platform.

ORNL-DWG 91-6254

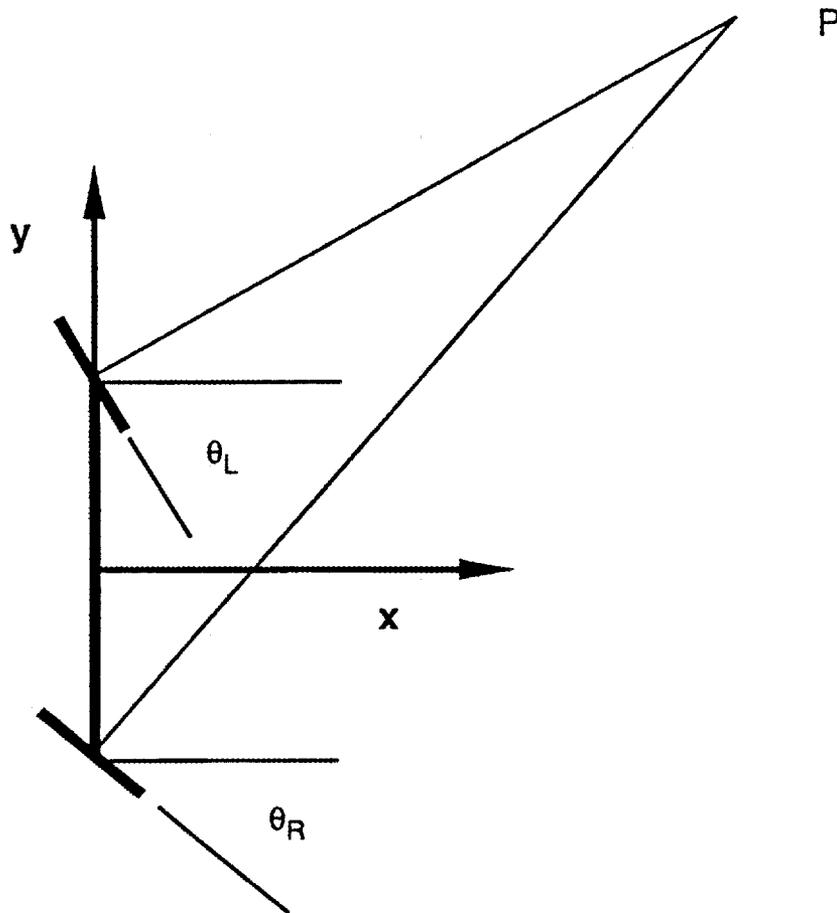


Fig. 5. The local coordinate system for the platform.

## 2.1 PLATFORM GOAL

The input interface is a platform goal. The goal for the robot could be traveling through a point at a speed; moving to a point, stopping, and rotating to a posture; or rotating about a point until reaching a final orientation. The robot could move with wheels pointed forward or could steer the wheels. As the platform rotates about an instantaneous center of rotation (ICR), the direction of the wheels is constant in the local coordinate system and variable in the world coordinate system. To specify the direction of the wheels, the user can specify the ICR for the robot. Thus, a platform goal contains a point  $(x,y)$ , a direction  $(\phi)$ , a speed  $(s)$ , and the ICR  $(P)$  [ $x,y,\phi$ , and  $s$  are defined in the world coordinates (Fig. 4), while  $P$  is defined in the local coordinates (Fig. 5)]. In addition, the platform goal contains a mode variable.

A user might want to move with wheels pointed forward through a point at maximum speed and might not care about the orientation of the robot while passing the point. The mode variable specifies which parts of the information in the platform goal can be ignored. The eight values for the mode variable are labeled in Table 1.

## 8 ARCHITECTURE

When the mode is zero, all information in the platform goal can be ignored and the appropriate response is to quit. For modes 1 to 3, the wheels point forward and the motions are called Fixed. For modes 4 to 7, the directions of the wheels can change and the motions are called Steer. For modes 1 to 7, the magnitude of the speed is always significant, while sometimes the sign of the speed is ignored.

**Table 1. Allowable Types of Motion Toward the Goal.**

Mode	Action by Robot	(x,y)	$\phi$	P
0	End Plan			
1	Fixed Move Point	X		
2	Fixed Move Rotate		X	
3	Fixed Move Posture	X	X	
4	Steer Move Circle			X
5	Steer Move Point	X		
6	Steer Move Rotate		X	X
7	Steer Move Posture	X	X	

When the mode is one, all information in the platform goal can be ignored except the point (x,y) and the speed. The objective is to move through the point at the given speed with fixed wheels at an arbitrary orientation. If the speed is negative, the robot backs through the point. The robot rotates to face the point and moves through the point.

When the mode is two, the significant information is the direction [ $\phi$ ]. The robot stops translating, rotates with fixed wheels until it reaches the direction specified in the goal, and stops at an arbitrary point. Since the robot rotates in the direction that will get to the goal in the shortest time at the given speed, the sign of the speed is ignored.

When the mode is three, both the point and direction are significant. The robot rotates to face the point, moves to the point, stops, and rotates to the given direction.

When the mode is four, the ICR (P) is the significant information. The robot steers the wheels to the two corresponding directions and rotates about the center of rotation at the given speed. When the mode is six, the robot stops when it reaches the desired orientation.

When the mode is five, the robot steers the wheels toward the goal and moves through the goal at the given speed.

When the mode is seven, the robot steers the wheels toward the goal, moves to the point, stops, straightens the wheels, and rotates to the given direction.

In modes one, four, and five, the robot does not stop. After passing the goal in modes one or five, the robot immediately attempts to pass through the goal again. In mode four, the robot rotates until a new goal is provided.

The speed is an upper bound on the speed of each wheel during a motion. Thus, changing the speed will affect both translation speed and rotation speed. In modes three and seven, the robot will automatically ramp down when it nears the goal. If the speed is zero in any mode, the robot will immediately ramp down to zero speed and will not reach the goal.

## 2.2 WHEEL TARGET AND CONTROL

A trajectory for the platform is determined by specifying a sequence of targets for the velocities and directions of the two wheels. We shall call the set of four numbers (two velocity and two direction) a wheel target. The HERMIES-II robot has limits on rotational speed and acceleration and on the steering speed and angle. For a typical trajectory, the wheels accelerate at the maximum rate until reaching the maximum speed, move at the maximum speed, and decelerate at the maximum rate to zero. Simultaneously, the direction of the wheels could be changing at the maximum rate. Thus, a trajectory can be specified by a sequence of values for the rotational accelerations and the steering velocities. Each set of values will have a length of time for their implementation. We shall call the set of five numbers (two acceleration, two velocity, and time) a wheel control. The current state of wheel velocity and direction plus the change caused by applying the wheel control for the specified time yields the next wheel target. In Fig. 3, the wheel controller determines the wheel targets and controls that are the inputs to the wheel driver.

## 2.3 WHEEL STATE

The wheel state is a data set that has the detailed joint variables for each wheel. For each wheel, the wheel state has data on eight variables: position, direction in world and local coordinates, rotation and velocity of the wheel about its axis, and the distance and angle from the ICR to the wheel. In addition, the wheel state has the local and world coordinates of the ICR.



### 3. WHEEL DRIVER AND RECKONER

This section describes the driver and reckoner. The inputs to the wheel driver are the wheel targets and controls and the outputs are the setpoints and rates for the motor driver. Conversion factors convert the wheel targets to setpoints and the wheel controls to rates. The conversion factors are from physical coordinates (meters, radians, and seconds) to encoder units (clicks and ticks). The targets for rotation are velocity, while the targets for steering are direction. The wheel control contains changes for the four targets and time. The new target is related to the current value by the relationship.

$$\text{new} = \text{current} + \text{change} * \text{time}.$$

The outputs from the motor driver are encoder values that can be converted to cumulative rotation and direction for the wheels. Thus, the encoders provide the values of the joint variables for the wheels. To calculate the Cartesian variables, the reckoner integrates the equations of motion for the platform. Next, we develop the equations of motion for the platform.

Our robot has two wheels that can be steered. The motion of each wheel is described by four variables:  $x$ ,  $y$ ,  $w$ , and  $\sigma$ . The robot has two coordinate systems: world (Fig. 4) and local (Fig. 5). The world coordinates  $x, y$  locate the point of contact for the wheel. The variable  $w$  is the cumulative rotation of the wheel about its axis. The variable  $\sigma$  is the direction of the wheel in world coordinates. The units for  $x$ ,  $y$ , and  $w$  are meters, while the units for  $\sigma$  are radians.

In developing the equations of motion for the robot, we will consider kinematics and neglect dynamics. We consider position, velocity, and acceleration and neglect mass, force, inertia, motor currents, and power supplies. The equations of motion for each wheel are:

$$\dot{x} = v \cos(\sigma)$$

$$\dot{y} = v \sin(\sigma)$$

where  $\dot{x}$  is the  $x$  component of the wheel velocity ( $v = \dot{w}$ ). The wheel direction in world frame ( $\sigma$ ), is the sum of the wheel direction in local coordinates ( $\theta$ ) and the direction of the robot in the world frame ( $\phi$ ):

$$\sigma = \phi + \theta.$$

The position, orientation, and motion of the robot are described by  $\phi$  and by five variables for each wheel ( $x$ ,  $y$ ,  $w$ ,  $v$ , and  $\theta$ ); a total of 11 state variables. The equations of motion are:

$$\dot{x}_R = v_R \cos(\theta_R + \phi) \tag{1}$$

$$\dot{y}_R = v_R \sin(\theta_R + \phi) \tag{2}$$

$$\dot{x}_L = v_L \cos(\theta_L + \phi) \tag{3}$$

$$\dot{y}_L = v_L \sin(\theta_L + \phi) \tag{4}$$

$$\dot{w}_R = v_R \quad (5)$$

$$\dot{w}_L = v_L \quad (6)$$

$$\dot{v}_R = u_1 \quad (7)$$

$$\dot{v}_L = u_2 \quad (8)$$

$$\dot{\theta}_R = u_3 \quad (9)$$

$$\dot{\theta}_L = u_4 \quad (10)$$

$$\dot{\phi} = (v_R \cos(\theta_R) - v_L \cos(\theta_L))/D \quad (11)$$

where  $D$  is the distance between the wheels and the subscripts  $R$  and  $L$  denote the right and left wheels.

The first four equations relate the Cartesian components of wheel velocity in the world frame to the wheel velocity in the joint space and the wheel direction in world frame. The four control variables are the rotational acceleration ( $u_1$  and  $u_2$ ) and the angular velocity ( $u_3$  and  $u_4$ ). The equation for the rotation of the robot [Eq. (11)] is derived in Appendix A. Specification of the initial conditions and the control variables uniquely determines the motion.

The Cartesian variables for the platform are position and orientation ( $x$ ,  $y$ , and  $\phi$  in Fig. 4). The position is the midpoint between the two wheels, while the orientation is the angle between the  $x$ -axis and a vector that is perpendicular to the line joining the two wheels. Given the locations of the two wheels, the position and orientation of the platform is uniquely determined and vice versa. The joint variables are the rotation and velocity of each wheel ( $w$  and  $v$ ) and the steering angle of each wheel in the local coordinate system ( $\theta$ ).

Since the platform of a mobile robot is a rigid body, the distance between the two wheels is fixed. When either or both of the wheels are not pointing forward, the velocities of the two wheels must satisfy the Rigid Body Constraint. If the two wheels point in different directions, the two lines that are the axes of rotation for the wheels intersect at the ICR. The velocities of the wheels are proportional to the distances from the ICR. The distance from the ICR to the right wheel is proportional to the sine of the left wheel direction in the robot frame ( $\theta_L$ ) and vice versa. Thus, the velocities of the wheels are inversely proportional to  $\sin(\theta)$ :

$$v_R/v_L = \sin(\theta_L) \sin(\theta_R)$$

Consequently, the Rigid Body Constraint is:

$$v_R \sin(\theta_R) = v_L \sin(\theta_L) \quad (12)$$

The Rigid Body Constraint is derived in Appendix A.

Whenever the wheels are in the forward position [ $\theta_R = \theta_L = 0$ ], there are no constraints on the rotational velocities ( $v_R$  and  $v_L$ ). As soon as one or both of the wheels is steered from the forward position, the constraint becomes active and discontinuous changes in velocity may be required to satisfy the constraint. A safe

way to satisfy the constraint is to have the same magnitude for both steering angles ( $\theta_R = \theta_L$  and  $v_R = v_L$  or  $\theta_R = -\theta_L$  and  $v_R = -v_L$ ).

During each cycle, the reckoner integrates the equations of motion using a fourth-order Runge-Kutta method function from the Numerical Recipes<sup>3</sup> collection (the functions are rk4 and rk45). The state variables should match the encoder values. For wheel rotation, the encoder values are proportional to  $w$ . By dividing the change in  $w$  by the time since the last encoder reading, we can calculate the average velocity ( $v$ ) during the interval.



## 4. WHEEL CONTROLLER

The inputs to the wheel controller are a platform goal and the wheel state, while the outputs are the platform posture and the wheel target and control. The controller operates at 20 Hz. Thus, the controller plans the next step on the path to the goal rather than the whole path. Since the goal can be changed at any time, the initial conditions are arbitrary.

Determination of the posture using the data in the wheel state requires a simple calculation. The heart of the controller is a function called QuickPlan that calls one of the seven functions that calculate the target and control for the seven modes displayed in Table 1.

To illustrate the structure of the seven functions, we will discuss the first one, Fixed Move Point. For Fixed Move Point, the objective is to move through the point at the given speed with the wheels fixed in the forward direction throughout the trajectory. If the speed is negative, the mobile robot backs through the point. The goal can be reached by a rotation followed by a translation. We begin with the algorithm for rotation.

The outputs of Fixed Move Point are the wheel accelerations. As a first draft algorithm, we could assume that the accelerations are proportional to the angular difference ( $\delta$ ) between the platform orientation ( $\phi$ ) and the goal ( $\phi_g$ ):

$$\delta = \phi - \phi_g$$

We assume that the accelerations on the two wheels are equal in magnitude and opposite in sign:

$$u_1 = -u_2 = u \quad , \quad (14)$$

and that the acceleration is proportional to the angular difference:

$$u = -k\delta \quad (15)$$

where  $k$  is a positive constant. For fixed wheels  $\theta_R = \theta_L = 0$ , and Eq. (11) may be written:

$$\dot{\phi} = (v_R - v_L)/D \quad . \quad (16)$$

Using Eqs. (7) and (8):

$$\ddot{\delta} = -(2k/D)\delta \quad (17)$$

Thus, this algorithm yields the equation for a harmonic oscillator which has the undesirable property that the oscillations are undamped. To avoid undamped oscillations, we will use the angular difference to establish velocity targets.

The wheel velocities have two components: translational and rotational (given velocities for the two wheels, the translational component is the average of the two velocities while the rotational component is one half the difference between the two velocities). As a second draft algorithm, we could assume that the rotational velocity ( $v_r$ ) is proportional to the angular difference:

$$v_r = (v_R - v_L)/2 \quad (18)$$

$$v_r = -k\delta \quad (19)$$

## 16 WHEEL CONTROLLER

Using Eq. (16):

$$\dot{\delta} = -(2k/D)\delta \quad (20)$$

This algorithm will converge but it can be slow.

To improve the speed, we could use a bang-bang algorithm (during the motion, the wheels always accelerate (or decelerate) at their maximum rate). For a bang-bang algorithm, the velocity targets for the wheels ( $V_R$  and  $V_L$ ) have opposite signs during the rotation and the same sign during the translation. The key parameter in the algorithm is the switch time: the time to switch from rotation to translation. Furthermore, the algorithm is running at 20 Hz. On every cycle for arbitrary initial conditions (the goal and the current velocities), the algorithm must choose between translation and rotation.

On every cycle, we calculate the ramp down angle, the change in angle that would occur if the robot switched from its current velocity to pure translation toward the goal at the maximum rate. We add this ramp down angle to the current heading and compare the resultant angle to the goal. If the platform is heading to the right of the goal, the target velocities are  $V_R = s$  and  $V_L = -s$  (we assume that the speed ( $s$ ) is positive). If the platform is heading to the left of the goal, the target velocities are reversed. Having calculated target velocities for the two wheels, the accelerations are proportional to the differences between the current velocities and the target velocities.

The initial wheel velocities ( $v_R$  and  $v_L$ ) are arbitrary. The time ( $t_R$  and  $t_L$ ) required to move from the current velocity to the target velocity ( $s$ ) at the maximum acceleration ( $u$ ) is:

$$t_R = |v_R - s|/u \quad (21)$$

$$t_L = |v_L - s|/u \quad (22)$$

To coordinate the motion, both transitions should require the same time. Let  $T$  be the maximum of  $t_R$  and  $t_L$ . The distance ( $w_R$  and  $w_L$ ) traveled during the transition is:

$$w_R = (v_R + s)T/2 \quad (23)$$

$$w_L = (v_L + s)T/2 \quad (24)$$

The ramp down angle, the rotation ( $\phi_d$ ) during the transition is:

$$\phi_d = (w_R - w_L)/D \quad , \quad (25)$$

where  $D$  is the distance between the wheels.

Let  $\epsilon$  be the difference between the platform orientation after rampdown ( $\phi + \phi_d$ ) and the goal  $\phi_g$ :

$$\epsilon = \phi + \phi_d - \phi_g \quad (26)$$

If  $\epsilon > 0$ , the targets are  $V_R = -s$  and  $V_L = s$  (when  $s > 0$ ). If  $\epsilon < 0$ , the targets are reversed.

We use the bang-bang algorithm when the angular difference ( $\epsilon$ ) is large and the second algorithm [Eq. (19)] when the angular difference is small.

## 5. EXPERIMENTAL RESULTS

The motion system for the HERMIES-III robot (controller, driver, and reckoner) has been constructed and tested. Time delays have been added to prevent the cycle rate from being faster than the design goal of 20 Hz. To test the system, tools have been created to set platform goals for the system. The results of two typical experiments will be discussed in this section.

The first experiment was performed using fixed wheels in modes one and three. As shown in Fig. 6, the robot was initially at rest at the point (6,4) with a direction of 90 degrees (the units of x and y are meters). The initial goal for the robot was Fixed Move Point to (6,8). While the robot was moving forward, the goal was changed to Fixed Move Posture to (9,7) with a final direction of 0 degrees. The robot moved forward at full speed and then made a right turn and stopped at (9,7).

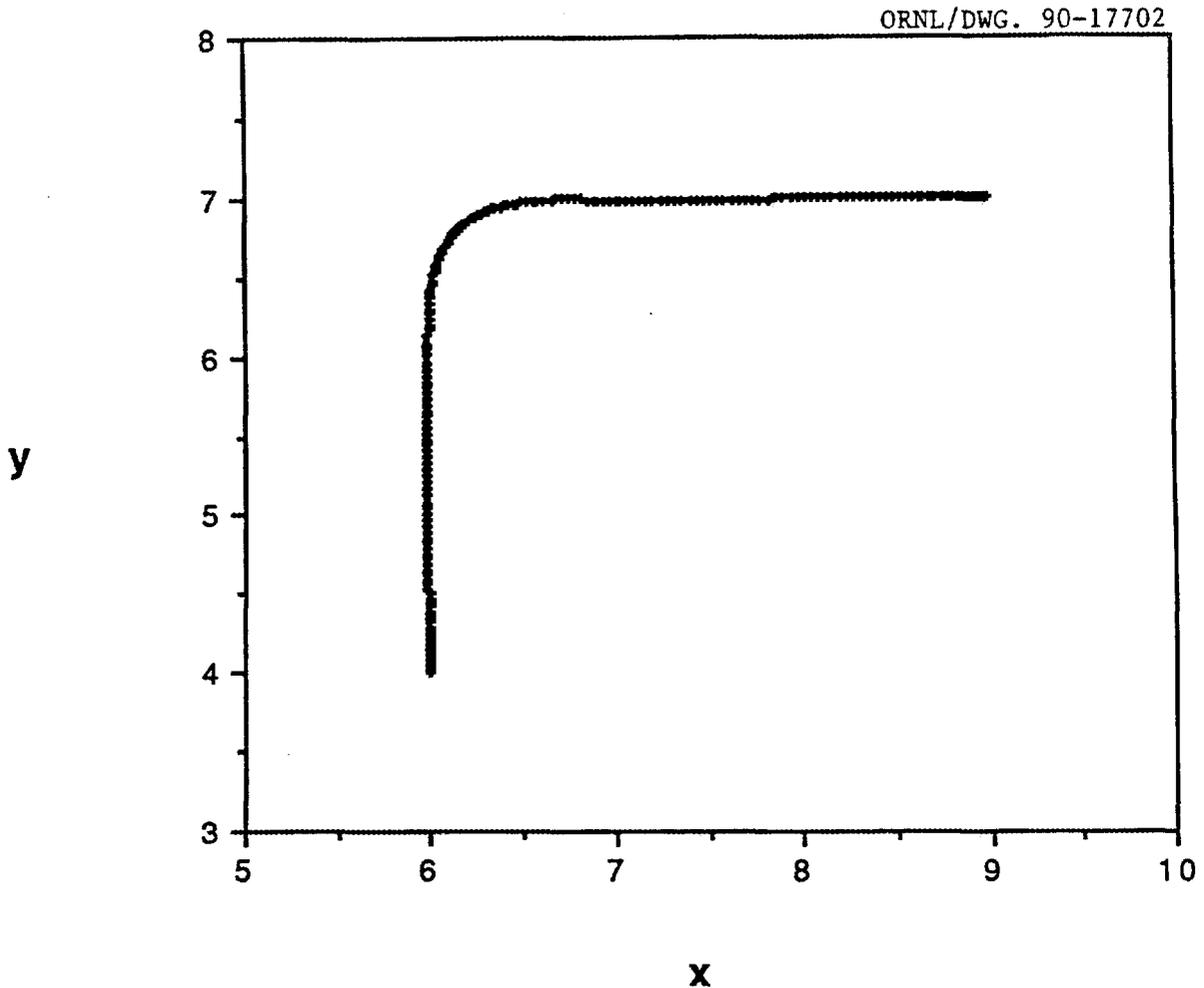


Fig. 6. Path of the robot during the first experiment. The units of x and y are meters.

## 18 EXPERIMENTAL RESULTS

The data displayed in Fig. 6 and the remaining figures of this section were collected from the robot during the experiment. Each cycle (approximately 20 times per second) the robot saves key data elements that are downloaded from memory to disk after the experiment.

For the first experiment, the left wheel accelerated to full speed (0.45 meters per second) and remained near full speed until it ramped down at the end of the experiment (see Fig. 7). The dip in speed at about 38 seconds was a final correction to lock in on the goal. Most of the steering was accomplished by the right wheel (see Fig. 8). For forward motion, both wheels accelerated to maximum speed. To make the right turn, the right wheel ramped down to zero and back up to top speed.

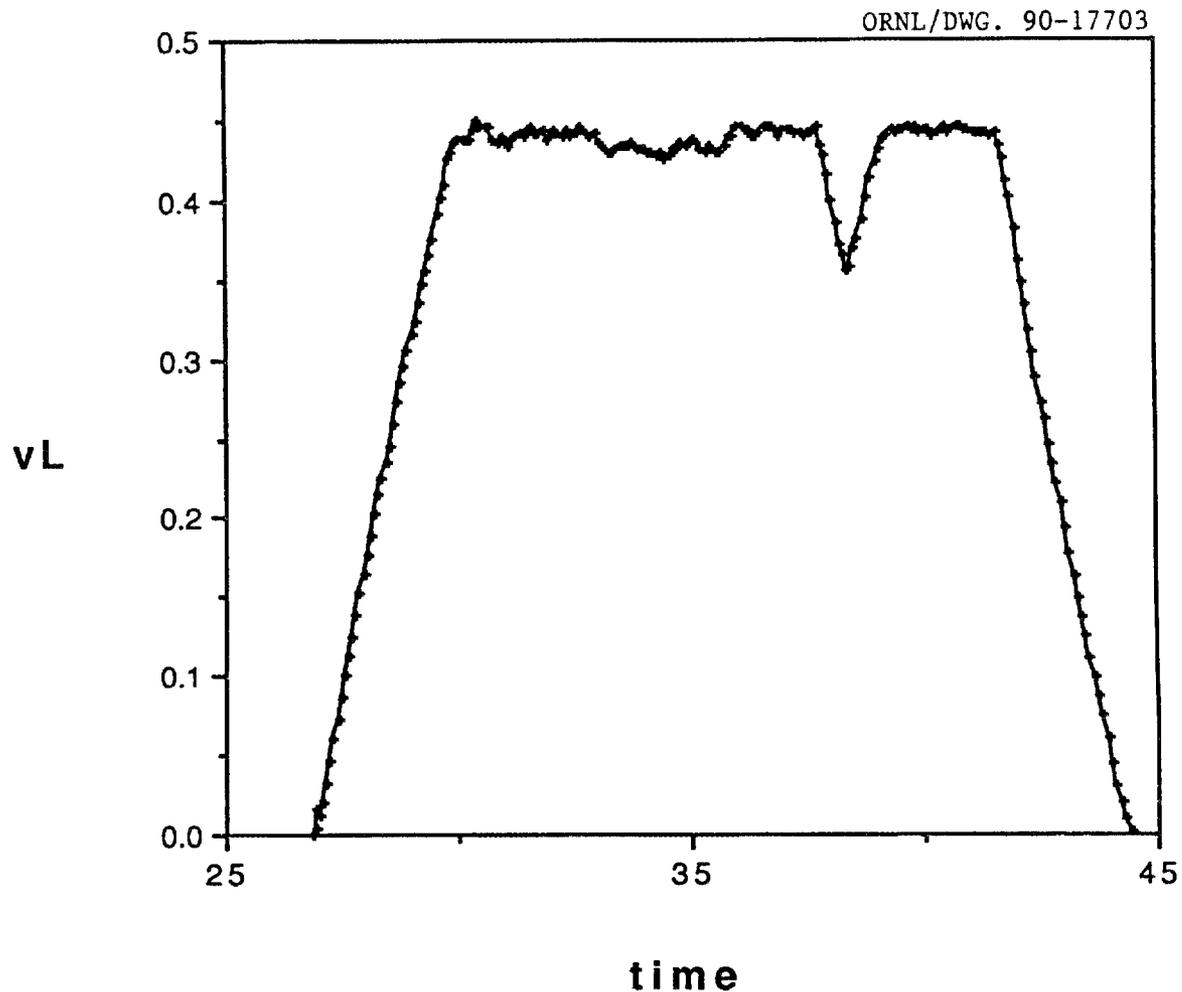


Fig. 7. The velocity of the left wheel during the first experiment ( $v_L = v_L$ ). The units of  $v_L$  are meters/second and the units of time are seconds.

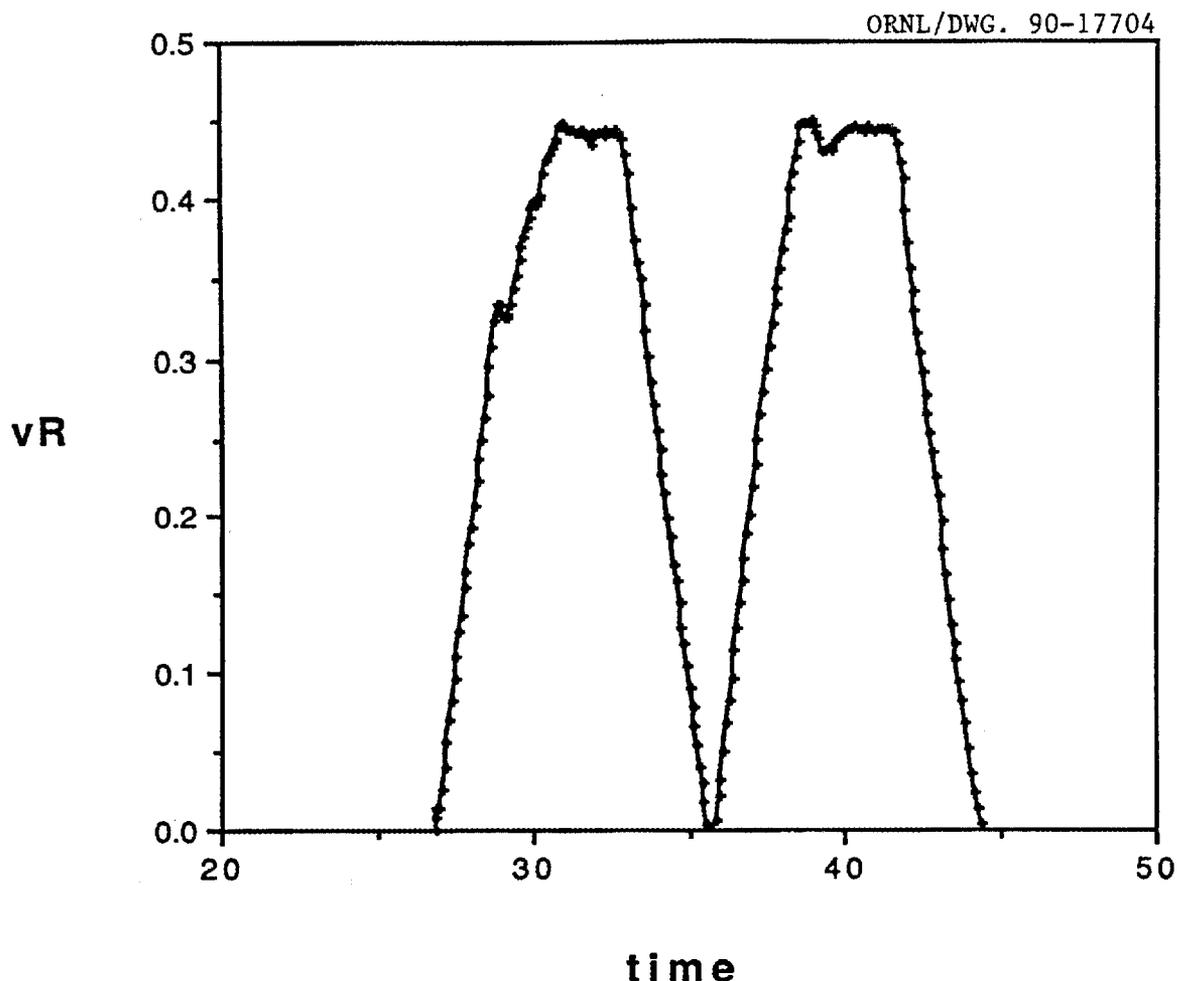


Fig. 8. The velocity of the right wheel during the first experiment ( $v_R = \dot{v}_R$ ). The units of  $v_R$  are meters/second and the units of time are seconds.

Target velocities and realized velocities for the two wheels during the initial acceleration are displayed in Figs. 9 and 10. For the left wheel, the realized velocity always trails the target (see Fig. 9). The targets for the right wheel are similar to the left wheel until they dip between 28 and 29 seconds (see Fig. 10). The data illustrate how quickly and smoothly the robot responds to changes in the target.

For the second experiment, the robot moved in mode five (Steer Move Point). As shown in Fig. 11, the robot was initially at rest at the point (6,4) with a direction of 90 degrees. At the start of the experiment, the goal was (8,6). To reach the goal, the robot ramped up to full speed and simultaneously steered the wheels toward the goal. After four seconds, the goal was changed to (5,7). In response, the robot smoothly changed its direction and continued to ramp up. After four seconds, the goal was changed to (8,8). At full speed, the robot changed direction. After four seconds, the goal was changed to (5,9) and the speed goal was reduced to zero. In response, the robot simultaneously changed direction and stopped (when the speed goal is zero, the robot cannot reach the goal (5,9)).

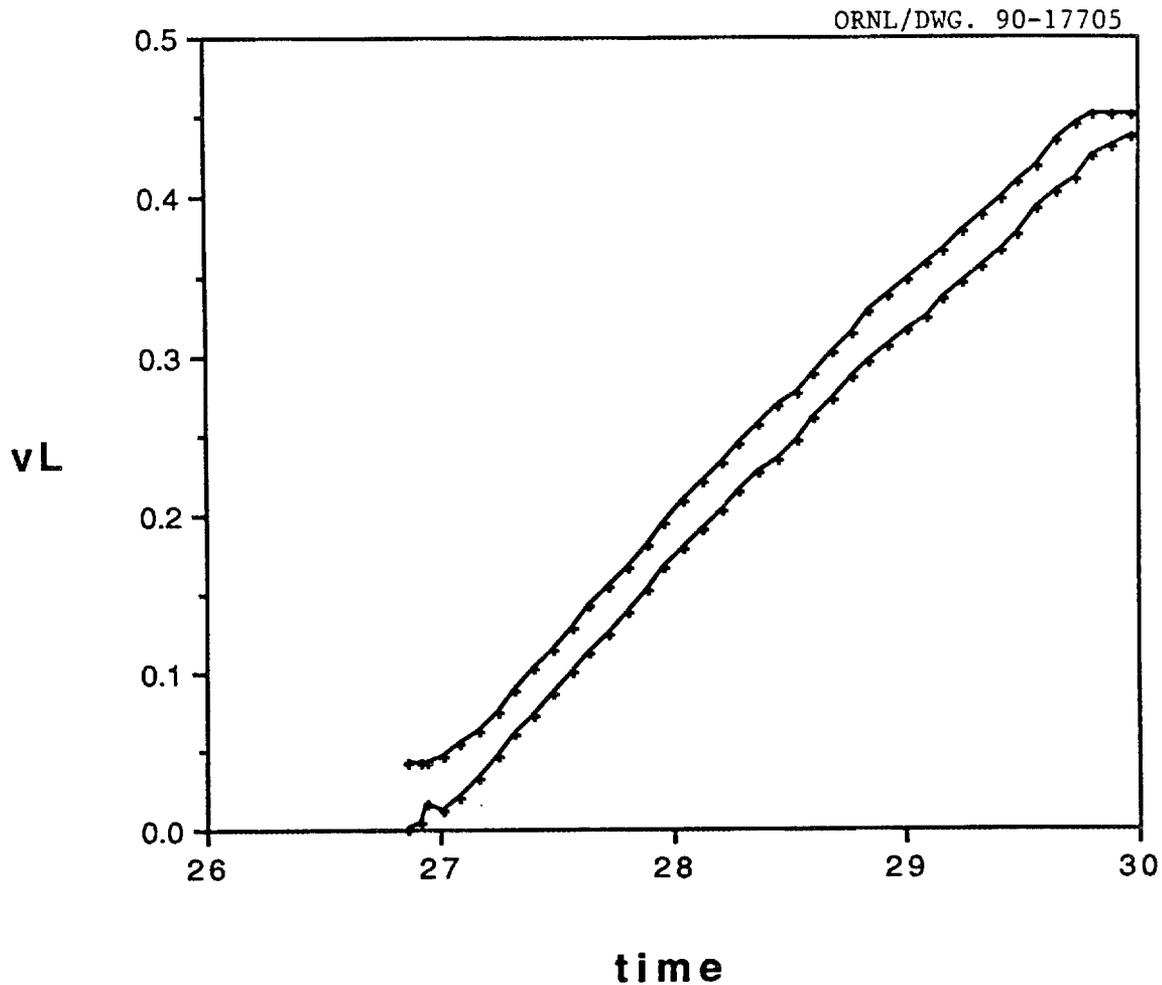


Fig. 9. The target velocity (upper) and realized velocity (lower) for the left wheel.

The direction of the right wheel during the second experiment is displayed in Fig. 12. During the experiment, the goal point was changed four times. In response, both of the wheels turned to the right, to the left, back to the right, and finally back to the left. Although sudden changes in steering velocity could have caused the steering angle to overshoot the goal, the data in Fig. 12 do not show overshoot.

The velocity of the right wheel is presented in Fig. 13. The velocity of both wheels ramp up to the maximum speed and remain at top speed until the ramp down at the end of the experiment.

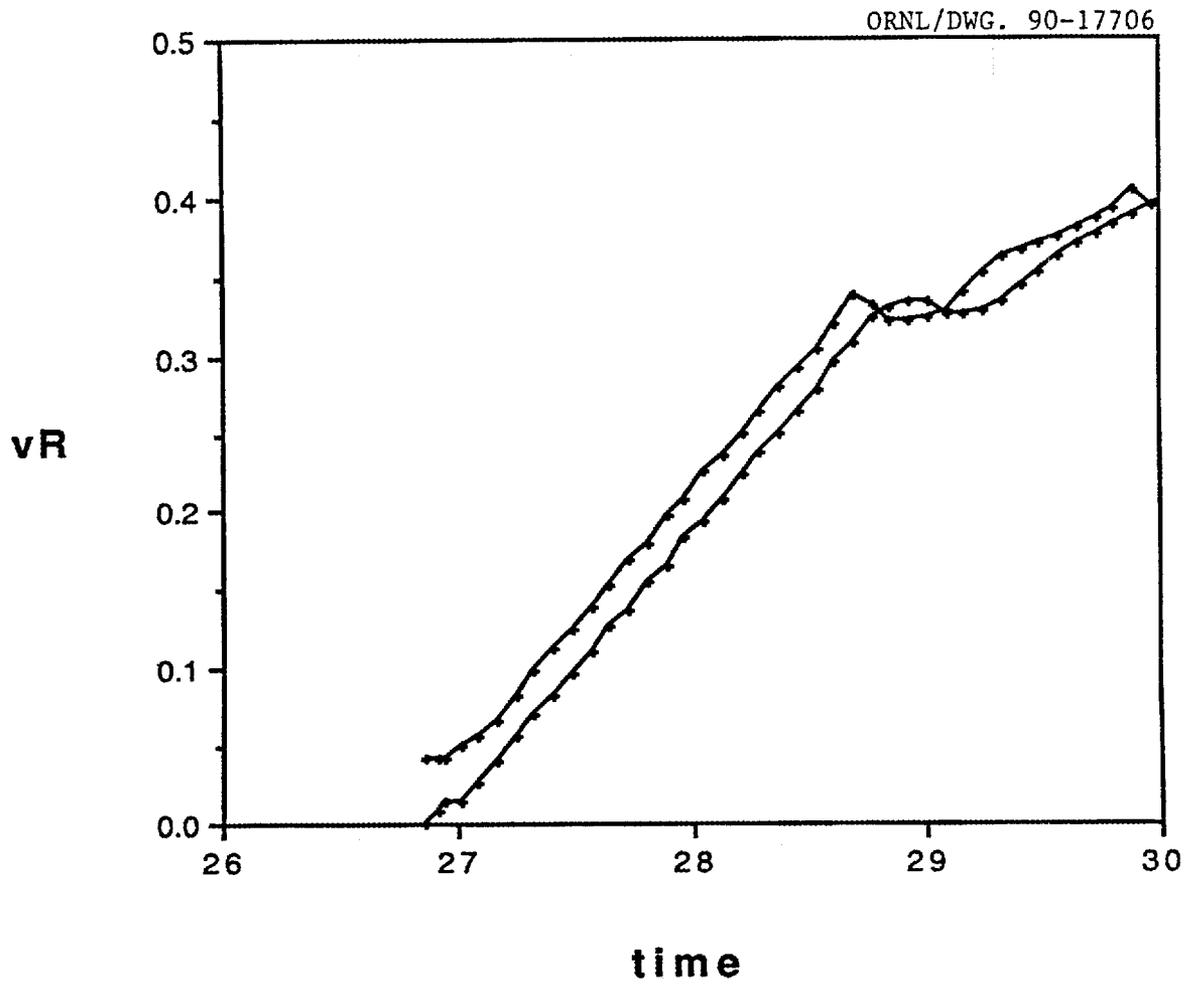


Fig. 10. The target (upper) and realized velocity (lower) for the right wheel.

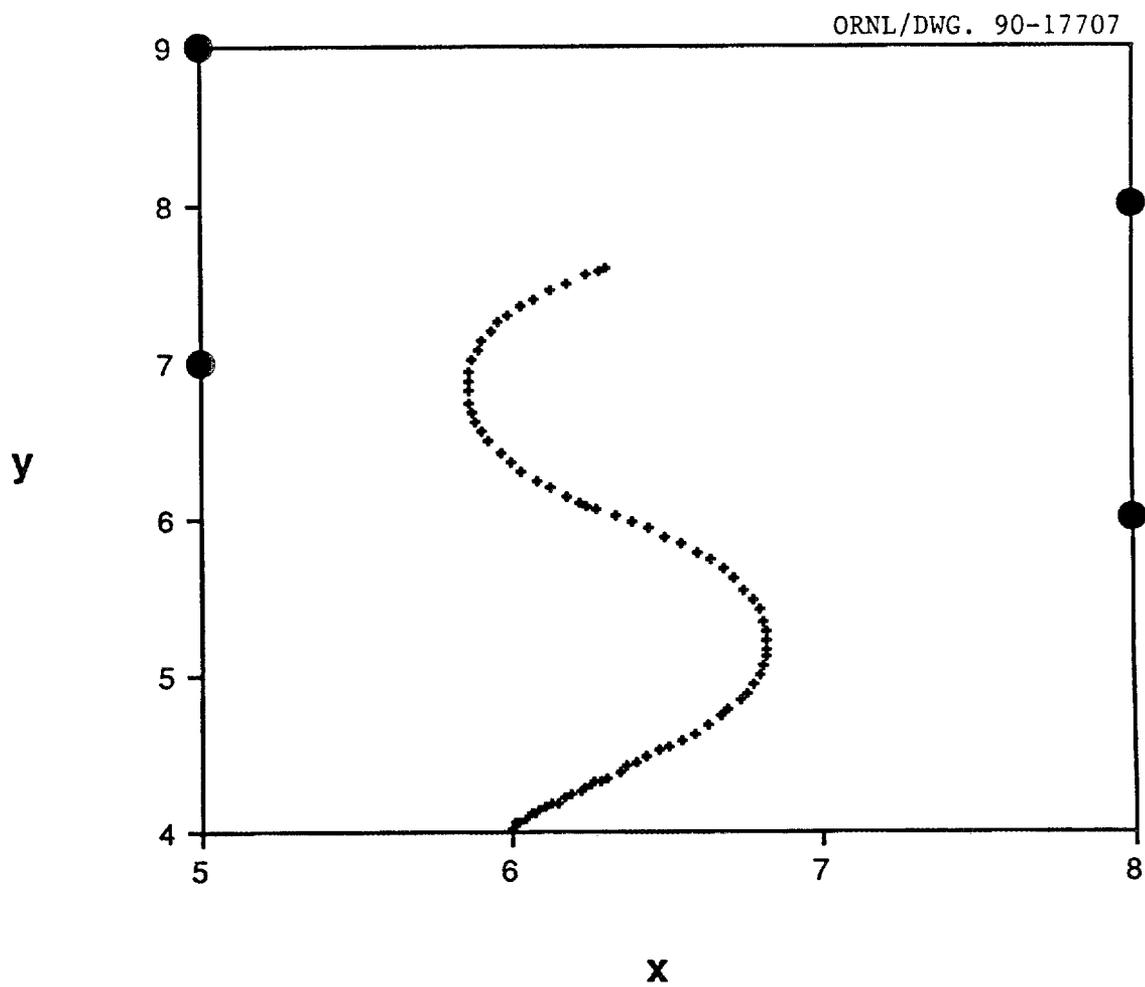


Fig. 11. Path of the robot during the second experiment. The units of x and y are meters.

ORNL-DWG 90-17708

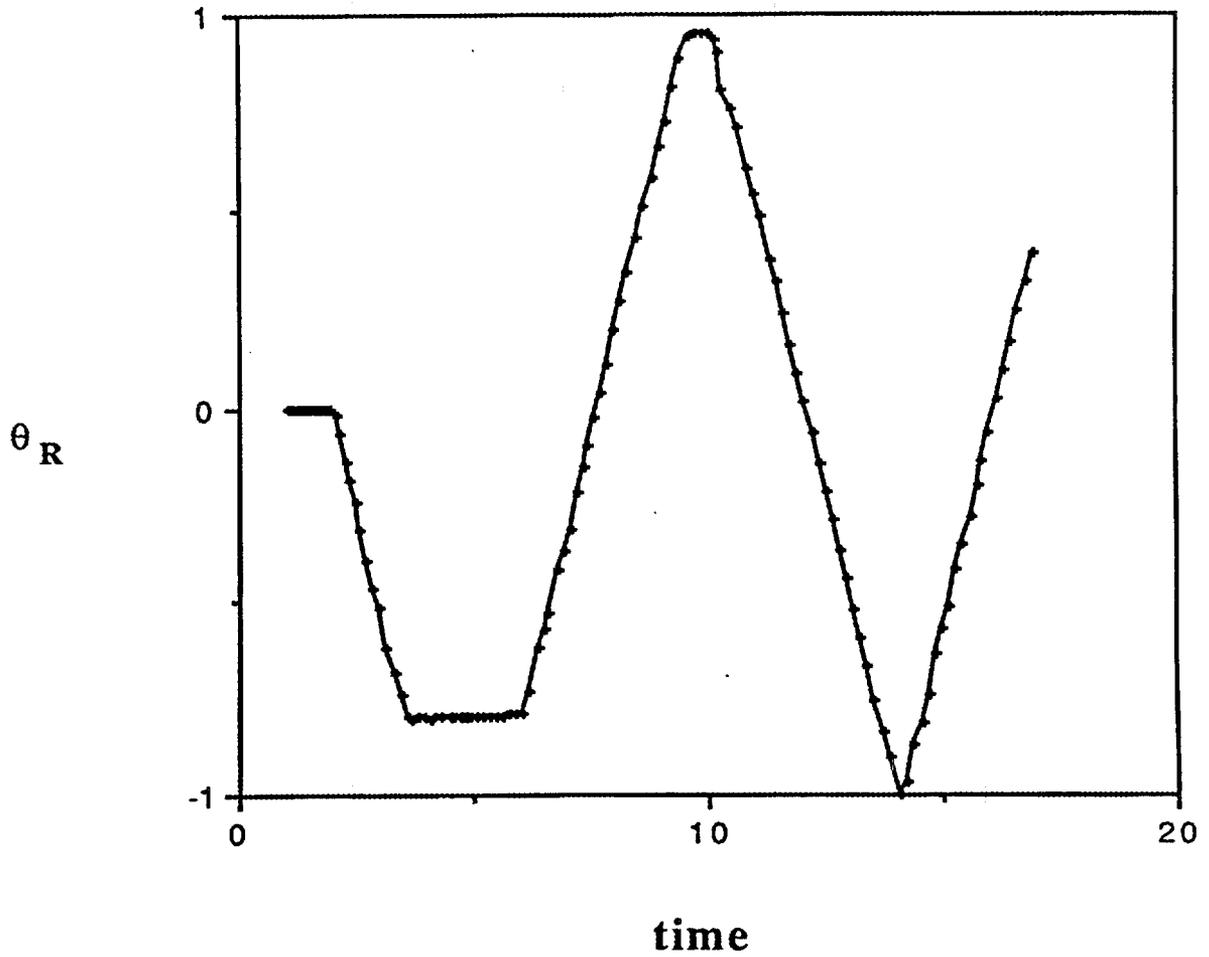


Fig. 12. Direction of the right wheel during the second experiment. The units of  $\theta_R$  are radians and the units of time are seconds.

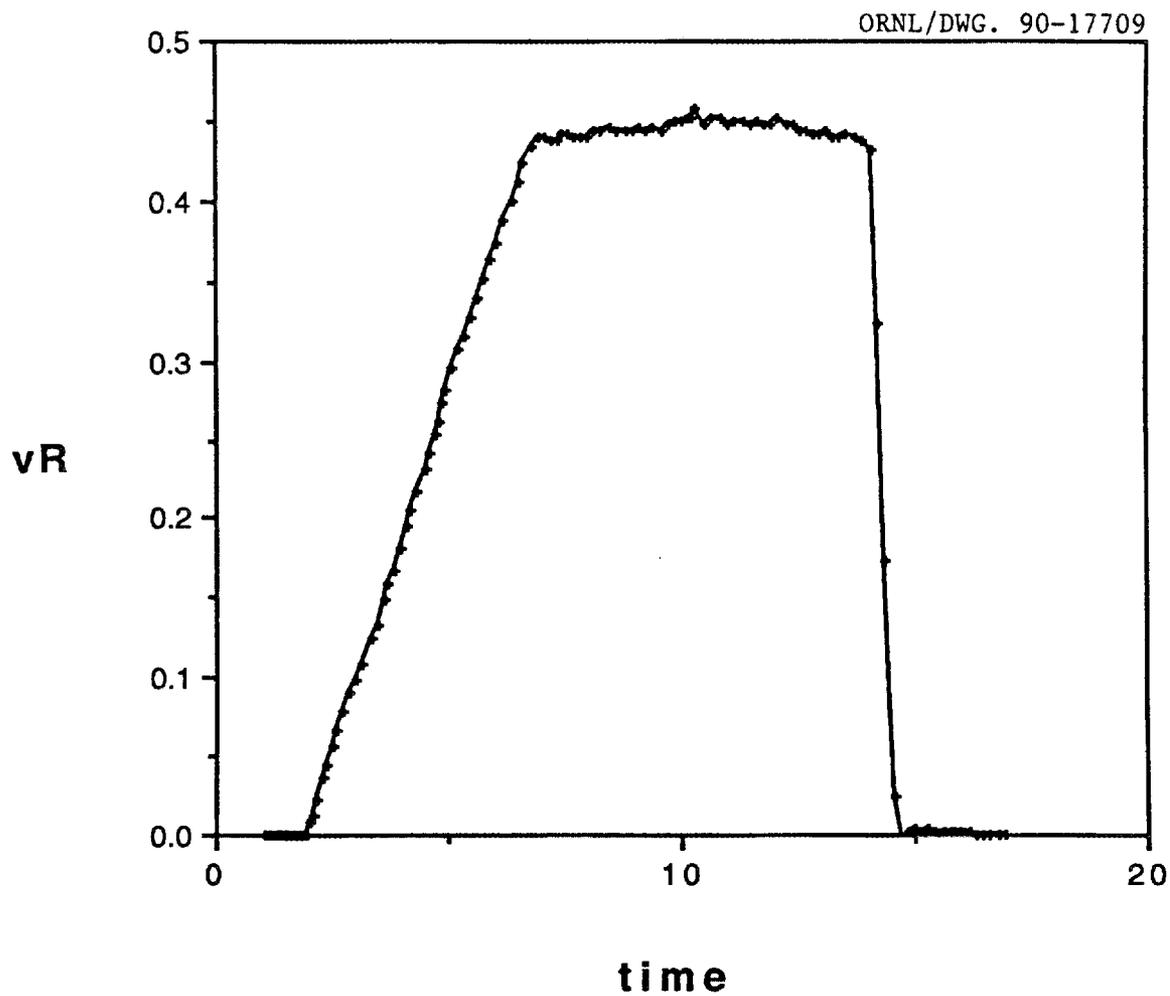


Fig. 13. The velocity of the right wheel during the second experiment ( $vR = v_R$ ). The units of  $vR$  are meters/second and the units of time are seconds.

## 6. CONCLUSIONS

We have designed, built, and tested a new wheel control system for the HERMIES-III robot. The system has seven modes for moving to a goal and allows the goal to be changed (at 20 Hz) during motion by the robot. This section will review some of the lessons that were learned during the construction of the software system.

The design of the wheel control system has evolved through several iterations. Initially, we planned to construct a system that would move to a goal from zero initial conditions. Since each movement could take several seconds, the rate at which goals could have been changed would have been slower than 1 Hz. We decided that the system must respond quickly to unexpected obstacles and chose a design objective of 20 Hz.

The wheel control system has three components. If each component is a separate process, the three processes need to be coordinated. If all of the components are in one process, coordination is guaranteed but the system is less flexible. We have decided to have two processes: one for the driver and reckoner and the second for the controller. In the future, alternate controllers (for example, a slower controller) could be used to send plans (target and control) to the driver.

We encountered coordination problems between the wheel driver and the motor driver. Our current solution is to delay for 0.04 seconds after a setpoint is sent to the motor driver. Future experimentation could change the delay time.

Initially, the controller calculated the wheel control and the wheel driver calculated the wheel target. However, we encountered coordination problems. The wheel state used by the controller to calculate the control could be for a different time than the wheel state in the driver used to calculate the target. Since the target is the sum of the control and the current values for the wheel state, we perform the addition in the controller and send the target to the driver.

The rigid body constraint relates velocities and wheel directions. Initially, we used values from the encoders to check the constraint and we shut down the system when the constraint was violated. Since the measured values can have a large variance, we were required to have large values for the tolerances to avoid frequent shutdowns. Next, we decided to use the values in the wheel target rather than measured values to check the constraint. However to test wheel slipping, we wanted to use targets that violated the rigid body constraint. Consequently, we are no longer shutting down the system when the constraint is violated.

Initially, we placed limits on acceleration for both ramp up and ramp down. Because of an unstable algorithm, we experienced a situation with velocity targets growing in magnitude and oscillating about the velocity limit at 20 Hz. The limit on deceleration prevented the velocity targets from reaching zero. Consequently we removed the limit on deceleration. We decided that a mobile robot might have limits on speed increases but should not have a limit on braking.

Rotation about an ICR can result in a substantial error; for example, when the robot concludes a 10 degree rotation about the ICR (2,2), the actual rotation can be 8 degrees (an error of -20%). We have spent several months performing experiments, formulating hypotheses, and eliminating potential sources of error. However, we have not been able to eliminate the error.

Our current hypothesis is that the error is caused by slipping wheels. When the ICR is fixed during an experiment (a 10 degree rotation), the rigid body constraint [Eq. (12)] requires a fixed ratio for the wheel velocities and the same fixed ratio for the distance traveled by each wheel during the experiment. During our experiments,

## 26 CONCLUSIONS

the targets for the wheel velocities have the proper ratio however the resultant velocities and distances do not have the proper ratio. Although the encoders report that the distances traveled by the wheels are incorrect, the physical distance between the wheels cannot change. Thus, when the encoders report that the wheels travel the wrong distances, they must be slipping on the floor and Eq. (11) will incorrectly estimate the amount of rotation.

When the rigid body constraint is active, the wheels cannot be controlled independently. If a wheel is fast one cycle, it cannot compensate by being slow the next cycle. Slip will occur during both cycles.

Reducing the error is an area for future research. Kankaanranta and Koivo<sup>4</sup> have developed a method for controlling constrained robots. While they have applied the method to constrained manipulators, the method could be applied to constrained platforms. The method features both a primary controller to track the desired trajectory (position, velocity, and force) under ideal conditions and a secondary controller to compensate for small deviations from the ideal trajectory caused by inaccurate modeling and disturbances.

## ACKNOWLEDGMENTS

This work was sponsored by the Engineering Research Program of the Office of Basic Energy Sciences and the Office Nuclear Energy, Office of Technology Support Programs of the U.S. Department of Energy under contract No. DE-AC05-84OR21400.



## REFERENCES

1. C. R. Weisbin, et al., "HERMIES-III: A Step Toward Autonomous Mobility, Manipulation, and Perception," *Robotica* **8**, 7-12 (1990).
2. J. F. Jansen and R. L. Kress, "Analysis and Control of a Three-Degree-of-Freedom Robot Platform," *Journal of Mechanical Working Technology* **20**, 295-304 (1989).
3. W. H. Press, et al., "Numerical Recipes in C: The Art of Scientific Computing," Cambridge University Press, Cambridge, UK (1988).
4. R. K. Kankaanranta and H. N. Koivo, "Dynamics and Simulation of Compliant Motion of a Manipulator," *IEEE Journal of Robotics and Automation*, **4**, 163-173 (1988).



## APPENDIX A

### KINEMATICS

This appendix derives the equation for the rotation of the robot and derives the Rigid Body Constraint.

We define  $F$  to be the vector from the left wheel to the right wheel:

$$F_x = x_R - x_L \quad (1)$$

$$F_y = y_R - y_L \quad (2)$$

Since the direction of the robot is perpendicular to  $F$  (see Fig. 2):

$$F_x = D \sin(\phi) \quad (3)$$

$$F_y = -D \cos(\phi) \quad (4)$$

If we take the time derivatives of Eqs. (3) and (4):

$$\dot{F}_x = D \cos(\phi) \dot{\phi} \quad (5)$$

$$\dot{F}_y = D \sin(\phi) \dot{\phi} \quad (6)$$

If we take the time derivatives of Eqs. (1) and (2) and use the equations of motion:

$$\dot{F}_x = v_R \cos(\theta_R + \phi) - v_L \cos(\theta_L + \phi) \quad (7)$$

$$\dot{F}_y = v_R \sin(\theta_R + \phi) - v_L \sin(\theta_L + \phi) \quad (8)$$

Using the expressions for the sin and cos of the sum of two angles, we can derive two equations for  $\dot{\phi}$ . To compact our equations, we will use the notation:  $s_R = \sin(\theta_R)$  and  $c_L = \cos(\theta_L)$ . Using Eqs. (5) to (8):

$$D \dot{\phi} = v_R c_R - v_L c_L - (v_R s_R - v_L s_L) * \tan(\phi) \quad (9)$$

$$D \dot{\phi} = v_R c_R - v_L c_L + (v_R s_R - v_L s_L) / \tan(\phi) \quad (10)$$

Since the left sides of Eqs. (9) and (10) are identical, the right sides must be equal. Thus,

$$\dot{\phi} = (v_R c_R - v_L c_L) / D \quad (11)$$

and

$$v_R s_R = v_L s_L \quad (12)$$

Equation (11) is the equation for the rotation of the robot and Eq. (12) is the Rigid Body Constraint.



## APPENDIX B

### CONTROL ALGORITHMS

The wheel control system has seven modes. In each mode, the wheel controller plans the next step on the path to the goal. In this appendix, we will derive several of the algorithms that are used by the controller.

#### FIXED MOVE POINT

The objective is to move through a point at a given speed. The robot rotates and then translates. During each cycle, the controller must decide whether to be rotating (with opposite wheel velocities) or to be translating (with equal wheel velocities). The control algorithm calculates how far the robot will rotate during the transition from rotation to translation.

The initial wheel velocities ( $v_R$  and  $v_L$ ) are arbitrary. The time ( $t_R$  and  $t_L$ ) required to move from the current velocity to the target velocity ( $s$ ) at the maximum acceleration ( $u$ ) is:

$$t_R = |v_R - s|/u \quad (1)$$

$$t_L = |v_L - s|/u \quad (2)$$

To coordinate the motion, both transitions should require the same time. Let  $T$  be the maximum of  $t_R$  and  $t_L$ .

The distance ( $w_R$  and  $w_L$ ) traveled during the transition is:

$$w_R = (v_R + s)T/2 \quad (3)$$

$$w_L = (v_L + s)T/2 \quad (4)$$

The rotation ( $\phi$ ) during the transition is:

$$\phi = (w_R - w_L)/D \quad (5)$$

where  $D$  is the distance between the wheels.

#### FIXED MOVE ROTATE

The objective is to rotate until reaching the direction specified in the goal. The controller calculates the rotation during the ramp down to zero velocity using Eq. (5) with  $s = 0$ . After the ramp down, the robot may be slightly off the specified direction. During the end game, the robot ramps up (to  $v$ ) one time step ( $t$ ) and ramps back to zero the next time step. The distance traveled by the right wheel during the ramp up and down is  $w_R = vt$ . The distance traveled by the left wheel is  $w_L = -w_R$ . Using Eq. (5), the rotation ( $\phi$ ) is:

$$\phi = 2vt/D \quad (6)$$

Given the angle between the robot and the goal ( $\phi$ ) and the time step, Eq. (6) can be used to calculate the target velocity.

### FIXED MOVE POSTURE

The objective is to move to a point, stop, and rotate to an angle. The new feature is the ramp down required to stop at the point. Let  $d$  be the distance to the goal minus the distance traveled in one time step at the current speed. The distance traveled during a ramp down with deceleration ( $u$ ) for time ( $t$ ) is:

$$d = 0.5ut^2 \quad (7)$$

Since the target velocity ( $v$ ) is  $v = ut$ , Eq. (7) may be written:

$$d = 0.5v^2/u \quad (8)$$

Given  $d$  and  $u$ , Eq. (8) can be used to calculate the target velocity. The robot moves at the speed specified in the goal until the target velocity requires it to ramp down.

### STEER MOVE ROTATE

The objective is to rotate about the given center of rotation until reaching the goal direction. The controller calculates the rotation during the ramp down to zero velocity and the velocity for the end game. Since the steering angles are arbitrary, our previous results [Eqs. (5) and (6)] will be modified to include the steering angles  $\theta_R$  and  $\theta_L$ . We will assume that the steering angles are constant. To compact our equations, we will use the notation:  $s_R = \sin(\theta_R)$  and  $c_L = \cos(\theta_L)$ .

The initial wheel velocities ( $v_R$  and  $v_L$ ) are arbitrary. The time ( $t_R$  and  $t_L$ ) required to move from the current velocity to the target velocity ( $s = 0$ ) at the maximum acceleration ( $u$ ) is given by Eqs. (1) and (2). The Rigid Body Constraint requires that both transitions should be coordinated. Let  $T$  be the maximum of  $t_R$  and  $t_L$ .

The distance ( $w_R$  and  $w_L$ ) traveled during the transition is given by Eqs. (3) and (4). The rotation ( $\phi$ ) during the transition is:

$$\phi = (w_R c_R - w_L c_L)/D \quad (9)$$

During the end game, the robot ramps up (to  $v_R$  and  $v_L$ ) one time step ( $t$ ) and ramps back to zero the next time step. The distance traveled by the right wheel during the ramp up and down is  $w_R = v_R t$ . The distance traveled by the left wheel is  $w_L = v_L t$ . Using Eq. (9), the rotation ( $\phi$ ) is:

$$\phi = (v_R c_R - v_L c_L)t/D \quad (10)$$

During the end game, the velocities satisfy the Rigid Body Constraint:

$$v_R s_R = v_L s_L \quad (11)$$

Given the angle between the robot and the goal ( $\phi$ ), the steering angles, and the time step ( $t$ ), Eqs. (10) and (11) can be used to calculate the target velocities ( $v_R$  and  $v_L$ ).

## INTERNAL DISTRIBUTION

- |                   |                              |
|-------------------|------------------------------|
| 1. B. R. Appleton | 18-22. R. C. Mann            |
| 2. J. E. Baker    | 23-27. F. G. Pin             |
| 3. A. L. Bangs    | 28-32. D. B. Reister         |
| 4. M. Beckerman   | 33. J. C. Schryver           |
| 5. B. L. Burks    | 34. P. F. Spelt              |
| 6. R. J. Carter   | 35. F. J. Sweeney            |
| 7. J. R. Einstein | 36. M. A. Unseren            |
| 8. K. Fujimura    | 37-41. R. C. Ward            |
| 9. C. W. Glover   | 42-43. Laboratory Records    |
| 10. W. R. Hamel   | Department                   |
| 11. J. N. Herndon | 44. Laboratory Records,      |
| 12. J. F. Jansen  | ORNL-RC                      |
| 13. J. P. Jones   | 45. Document Reference       |
| 14. H. E. Knee    | Section                      |
| 15. R. L. Kress   | 46. Central Research Library |
| 16. G. Liepins    | 47. ORNL Patent Section      |
| 17. E. M. Oblow   |                              |

## EXTERNAL DISTRIBUTION

48. Dr. Peter Allen, Department of Computer Science, 450 Computer Science, Columbia University, New York, NY 10027
49. Mr. Harry Alter, Division of Advanced Technology Development, U.S. Department of Energy, Washington, DC 20545
50. Dr. Wayne Book, Department of Mechanical Engineering, J. S. Coon Building, Room 306, Georgia Institute of Technology, Atlanta, GA 30332
51. Dr. Johann Borenstein, The University of Michigan, 1101 Beal Ave., Ann Arbor, MI 48109-2110
52. Professor Roger W. Brockett, Wang Professor of Electrical Engineering and Computer Science, Division of Applied Sciences, Harvard University, Cambridge, MA 02138
53. Professor Carl Crane, 202 Nuclear Science Center, University of Florida, Gainesville, FL 32611
54. Professor John J. Dorning, Department of Nuclear Engineering and Physics, Thornton Hall, McCormick Rd., University of Virginia, Charlottesville, VA 22901
55. Dr. Steven Dubowsky, Massachusetts Institute of Technology, Building 3, Room 469A, 77 Massachusetts Ave., Cambridge, MA 02139
56. Mr. Ralph C. Gonzalez, Department of Electrical and Computer Engineering, The University of Tennessee, Knoxville, TN 37996-2100
57. Dr. Avi Kak, Department of Electrical Engineering, Purdue University, Northwestern Ave., Engineering Mall, Lafayette, IN 47907
58. Dr. James E. Leiss, 13013 Chestnut Oak Dr., Gaithersburg, MD 20878
59. Dr. Oscar P. Manley, Division of Engineering, Mathematical, and Geosciences, Office of Basic Energy Sciences, ER-15, U.S. Department of Energy - Germantown, Washington, DC 20545
60. Professor Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green St., Urbana, IL 61801

61. Dr. Wes Snyder, Center for Communications and Signal Processing, North Carolina State University, P.O. Box 7914, Raleigh, NC 27695-7914
62. Professor Delbert Tesar, Department of Mechanical Engineering, 26 San Jacinto, University of Texas, Austin, TX 78712
63. Professor M. A. Trivedi, Department of Electrical and Computer Engineering, The University of Tennessee, Knoxville, TN 37996-2100
64. Professor James S. Tulenko, 202 Nuclear Science Center, University of Florida, Gainesville, FL 32611
65. Professor David K. Wehe, Department of Nuclear Engineering, 2355 Bonisteel Blvd., The University of Michigan, Ann Arbor, MI 48109-2106
66. Professor Mary F. Wheeler, Department of Mathematical Sciences, Rice University, P.O. Box 1892, Houston, TX 77251
67. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations Office, P.O. Box 2001, Oak Ridge, TN 37831-8600
- 68-77. Office of Scientific Technical Information, P.O. Box 62, Oak Ridge, TN 37831