KINEMATIC MODELING FOR FEEDBACK CONTROL OF AN OMNIDIRECTIONAL WHEELED MOBILE ROBOT*

Patrick F. Muir and Charles P. Neuman

Department of Electrical and Computer Engineering
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We have introduced a methodology for the kinematic modeling of wheeled mobile robots[1-2]. In this paper, we apply our methodology to *Uranus*[3], an omnidirectional wheeled mobile robot which is being developed in the Robotics Institute of Carnegie Mellon University. We assign coordinate systems to specify the transformation matrices and write the kinematic equations-of-motion. We illustrate the *actuated inverse* and *sensed forward* solutions; i.e., the calculation of actuator velocities from robot velocities and robot velocities from sensed wheel velocities. We apply the actuated inverse and sensed forward solutions to the kinematic control of Uranus by: calculating in *real-time* the robot position from shaft encoder readings (i.e., dead reckoning); formulating an algorithm to detect wheel slippage; and developing an algorithm for feedback control.

1. Introduction

We have formulated a methodology for the kinematic modeling[1-2] of wheeled mobile robots (WMRs). In this paper, we apply our methodology to the kinematic modeling and control of *Uranus*[3], a WMR which is being constructed in the Robotics Institute of Carnegie Mellon University for research and development in control, computer vision, path planning and sonar sensing. Uranus provides omnidirectional mobility (i.e., independent and simultaneous X and Y translations and Z rotation) with a stable, rectangular wheelbase. Wheelchairs[4] which utilize this kinematic structure have already shown advantages over conventional designs. For example, the sideways X translational degree-of-freedom allows more maneuverability between closely spaced obstacles than turning the wheelchair in the direction of travel before translating.

In Section 2, we describe the kinematic structure of Uranus, omnidirectional wheels, and our control system hardware. Then, in Section 3, we develop the kinematic equations-of-motion by assigning coordinate systems, writing the transformation matrices and formulating the wheel Jacobian matrices. The Jacobian matrix of each of the four wheels resolves the robot body velocities as linear combinations of the wheel velocities. The four wheel equations-of-motion are then combined to form the composite robot equation-of-motion. We solve the composite robot equation-of-motion in Section 4. The actuated inverse solution (in Section 4.1) calculates the velocities of the actuators (i.e., wheel drive motors) from robot body velocities. The sensed forward solution (in Section 4.2) calculates the robot body velocities from the sensed wheel velocities (i.e., wheel shaft encoder measurements). In Section 5, we turn to kinematic control of Uranus. Dead-reckoning (discussed in Section 5.1), which is the real-time calculation of the robot position from the wheel shaft encoder measurements, plays the role of feedback in our kinematic control algorithm. If undesired wheel slip occurs, the dead reckoned robot position is erroneous. We then formulate in Section 5.2 an algorithm to detect wheel slippage. In Section 5.3, we design a kinematics-based feedback control algorithm for real-time implementation on Uranus. Finally, in Section 6, we highlight our kinematic controller design.

2. Uranus

Uranus[3] (pictured in Figure 1) has the kinematic structure of the Wheelon wheelchair[4]: four omnidirectional wheels mounted at the corners of a rectangular frame. For kinematic modeling, we sketch the wheelbase in Figure 2. Each omnidirectional wheel[5] (shown in Figure 3) consists of a wheel hub about which 12 rollers are mounted at the angle $\eta = 45°$ to the wheel orientation. Each omnidirectional wheel has three degrees-of-freedom (DOFs). One DOF is in the direction of the wheel orientation; the second DOF is in the direction of the roller rotation; and the third DOF is the rotation about the point-of-contact between the wheel and the floor. In Figures 2 and 3, the rollers on the underside of the wheel (i.e., the rollers touching the floor), rather than the rollers which are actually visible from a top view, are shown to facilitate kinematic analysis.

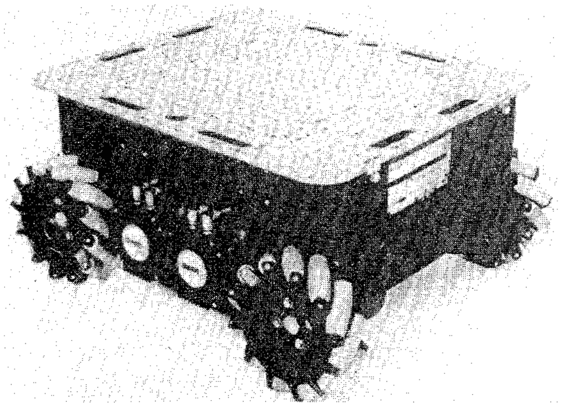

Figure 1: Uranus

The control system hardware consists of an onboard Motorola 68000 control processor, motors, shaft encoders and inter-

* This research has been supported by the Office of Naval Research under Contract N00014-81-K-0503 and the Department of Electrical and Computer Engineering, Carnegie Mellon University. The authors acknowledge Hans Moravec, Head of the Mobile Robot Laboratory, for his encouragement and support of this research.

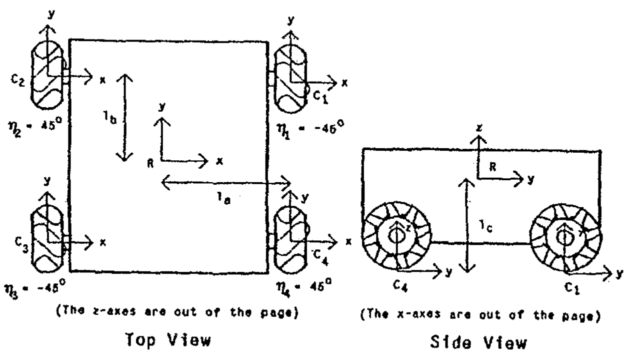

Figure 2: Coordinate System Assignments for Uranus

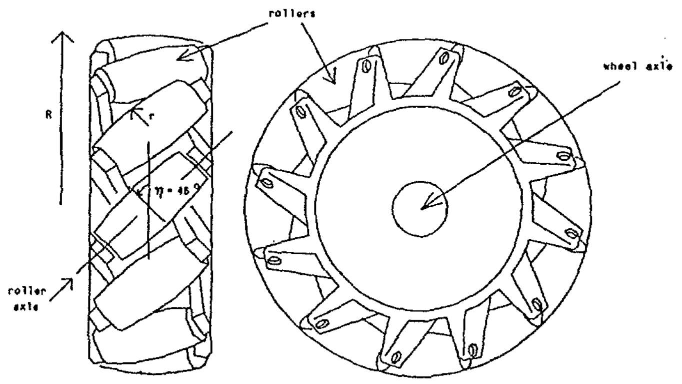

Figure 3: Omnidirectional Wheel

face electronics. The inner hub of each omnidirectional wheel is driven by a brushless DC motor and its position is sensed by an optical shaft encoder. The position and velocity of each wheel are thus available to the control processor through the interface electronics. The interface electronics also handle electronic commutation and pulse-width modulation for the brushless DC motors. Since the dynamic model of the motor velocity is *linear* in the pulse-width[6], the control processor need only calculate the wheel motor velocities. The desired robot position trajectory is communicated to the control processor from an independent offboard processor. A stiff suspension mechanism is built into each wheel assembly. The hardware construction of Uranus is nearing completion.

Three practical assumptions make the kinematic modeling of Uranus tractable. First, the motion of the suspension and compliance of the rollers are negligible. The assumption permits us to model the kinematics of the robot independently of the dynamics of these flexible components. Second, Uranus moves on a planar surface. We thus neglect irregularities in the floor surface. Even though this assumption restricts the range of practical applications, environments which do not satisfy this assumption (e.g., rough, bumpy or rocky surfaces) do not lend themselves to energy-efficient wheeled-vehicle travel[7]. Third, the rotational friction at the point-of-contact between a wheel and the surface is small enough to permit rotational slip. We make this assumption to treat the rotation about the point-of-contact (i.e., rotational wheel slippage) as a wheel DOF. Automobiles illustrate the practicality of this assumption since the wheels must rotate about their points-of-contact to navigate a turn.

3. Kinematic Modeling

3.1. Coordinate System Assignments

We apply the Sheth-Uicker convention[8] to assign coordinate systems. The Sheth-Uicker convention differs from the Denavit-Hartenburg[9] convention, which is applied to model stationary manipulators, because two coordinate systems are assigned at each joint (one in each link) instead of one. The displacement between the pair of coordinate systems at a joint is specified by the joint parameters. The Sheth-Uicker convention thereby allows the modeling of the higher-pair, multidimensional wheel motion and eliminates ambiguities in coordinate transformation matrices of closed-link chains.

Coordinate system assignment is illustrated in Figure 2 for Uranus. The stationary *floor* coordinate system F serves as the absolute reference coordinate frame for robot motions. The floor coordinate system is assigned with the z-axis orthogonal to the surface of travel. The *robot* coordinate system R is assigned to the robot body with the z-axis orthogonal to the surface of travel. The motion of the robot coordinate system is interpreted as the motion of the robot. We have assigned the robot coordinate system at the center of the robot body (l_a and l_b are one-half of the robot width and length, respectively), and at a height l_c above the floor. We assign a *contact point* coordinate system C_i (for $i = 1, 2, 3, 4$) at the point-of-contact between each wheel and the floor with the y-axis parallel to the wheel orientation, and the x-y plane tangent to the surface of travel.

For stationary serial link manipulators, all joints are one-dimensional lower-pairs: prismatic joints allow translational Z motion and revolute joints allow rotational θ motion. The velocity of a manipulator joint can be specified independently of its absolute position by referencing the velocity of the coordinate system in the link on one side of the joint to the coordinate system in the adjacent link on the other side of the joint. In contrast, WMRs have three-dimensional higher-pair wheel-to-floor and robot-to-floor joints allowing simultaneous X, Y and θ motions. Such multi-dimensional joints require the assignment of an *instantaneously coincident coordinate system*[1-2] to specify the velocity of the joint independently of its position. An instantaneously coincident coordinate system $\overline{A}$ for the coordinate system A is defined to be at the same position and orientation in space as the A coordinate system, but stationary relative to the absolute F coordinate system. For example, an *instantaneously coincident robot* coordinate system $\overline{R}$ is defined at the same position and orientation in space as the robot coordinate system R, but stationary relative to the F coordinate system. We may thus specify the velocities of the robot coordinate system R independently of the robot position by referencing the velocities of the R coordinate system to the $\overline{R}$ coordinate system (as in Section 3.3). When the robot coordinate system moves relative to the floor coordinate system, we assign a different instantaneously coincident robot coordinate system at each time instant. Similarly, the *instantaneously coincident contact point* coordinate system $\overline{C}_i$ (for $i = 1, 2, 3, 4$) coincides with the contact point coordinate system C_i and is stationary relative to the floor coordinate system. We require the instantaneously coincident contact point coordinate system to specify the wheel velocities independently of the wheel positions.

3.2. Transformation Matrices

Homogeneous (4×4) transformation matrices express the relative positions and orientations of coordinate systems[10]. The homogeneous transformation matrix ${}^{A}\Pi_{B}$ transforms the coordinates of the point ${}^B\mathbf{r}$ in the coordinate frame B to its corresponding coordinates ${}^A\mathbf{r}$ in the coordinate frame A:

$$ {}^A\mathbf{r} = {}^A\mathbf{\Pi}_B \; {}^B\mathbf{r} \; . \qquad (1)$$

We adopt the following notation. Scalar quantities are denoted by lower case letters (e.g., w). Vectors are denoted by lower case boldface letters (e.g., $\mathbf{r}$). Matrices are denoted by upper case boldface letters (e.g., $\mathbf{\Pi}$). Pre-superscripts denote reference coordinate systems. For example, ${}^A\mathbf{r}$ is the vector $\mathbf{r}$ in the A coordinate frame. Post-subscripts denote coordinate systems or components of a vector or matrix. For example, the transformation matrix ${}^A\mathbf{\Pi}_B$ defines the position and orientation of coordinate system B relative to coordinate frame A; and r_x is the x-component of the vector $\mathbf{r}$.

Our nomenclature for scalar rotational and translational displacements and velocities is:

${}^A\theta_B$: The rotational displacement about the z-axis of the A coordinate system between the x-axis of the A coordinate system and the x-axis of the B coordinate system (counterclockwise by convention); and ${}^A\dot{\theta}_B = {}^A\omega_B$.

θ_{w_ix} : The rotational displacement of wheel i (for $i = 1,2,3,4$) about an axis centered in the wheel and parallel to the x-axis of C_i; and $\dot{\theta}_{w_ix} = \omega_{w_ix}$.

θ_{w_ir} : The rotational displacement of the roller of wheel i (for $i = 1,2,3,4$) which is in contact with the floor about an axis aligned with the axle of the roller; and $\dot{\theta}_{w_ir} = \omega_{w_ir}$.

θ_{w_iz} : The rotational displacement of wheel i (for $i = 1,2,3,4$) about an axis centered in the wheel and parallel to the z-axis of C_i; and $\dot{\theta}_{w_iz} = \omega_{w_iz}$.

${}^Ad_{Bj}$: The translational displacement along the j-axis (for $j \in \{x,y,z\}$) of the A coordinate system between the origins of the A and B coordinate systems; and ${}^A\dot{d}_{Bj} = {}^Av_{Bj}$.

Since any two coordinate systems A and B in our WMR model are located at non-zero x, y and z-coordinates relative to each other, each transformation matrix must contain the translations ${}^Ad_{Bx}$, ${}^Ad_{By}$ and ${}^Ad_{Bz}$. We have assigned all coordinate systems with the z-axes perpendicular to the surface of travel so that all rotations between coordinate systems are about the z-axis. The transformation matrix ${}^A\mathbf{\Pi}_B$ thus embodies a rotation ${}^A\theta_B$ about the z-axis of coordinate system A, and the translations ${}^Ad_{Bx}$, ${}^Ad_{By}$ and ${}^Ad_{Bz}$ along the respective coordinate axes:

$$ {}^A\mathbf{\Pi}_B = \begin{pmatrix} \cos {}^A\theta_B & -\sin {}^A\theta_B & 0 & {}^Ad_{Bx} \\ \sin {}^A\theta_B & \cos {}^A\theta_B & 0 & {}^Ad_{By} \\ 0 & 0 & 1 & {}^Ad_{Bz} \\ 0 & 0 & 0 & 1 \end{pmatrix} \; . \qquad (2)$$

The pertinent coordinate transformation matrices are written for Uranus directly from Figure 2:

$$ {}^R\mathbf{\Pi}_{C_1} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} , \quad {}^R\mathbf{\Pi}_{C_2} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$ {}^R\mathbf{\Pi}_{C_3} = \begin{pmatrix} 1 & 0 & 0 & -l_a \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} , \quad {}^R\mathbf{\Pi}_{C_4} = \begin{pmatrix} 1 & 0 & 0 & l_a \\ 0 & 1 & 0 & -l_b \\ 0 & 0 & 1 & -l_c \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

Only one coordinate transformation matrix is written for each wheel of Uranus. More generally, wheels with steering links require the assignment of multiple coordinate systems[1-2]. The *sparse* transformation matrices result from assigning the robot coordinate system parallel to the wheel coordinate systems. Our robot coordinate system assignment thus facilitates kinematic modeling and control.

3.3. Wheel Jacobian Matrices

The Jacobian matrix $\mathbf{J}_i$ for wheel i resolves the robot velocities ${}^{\overline{R}}\dot{\mathbf{p}}_R$ as linear combinations of the wheel velocities $\dot{\mathbf{q}}_i$:

$$ {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_i \dot{\mathbf{q}}_i \qquad \textit{for } i = 1,2,3,4 \; . \qquad (3)$$

The robot velocity vector ${}^{\overline{R}}\dot{\mathbf{p}}_R$ consists of three components: the translational robot velocities ${}^{\overline{R}}v_{Rx}$ and ${}^{\overline{R}}v_{Ry}$, and the rotational robot velocity ${}^{\overline{R}}\omega_{Rz}$. The three DOFs of omnidirectional wheel i, wheel hub rotation, roller rotation, and rotation about the point-of-contact, correspond respectively to the three components ω_{w_ix}, ω_{w_ir}, and ω_{w_iz}, of the wheel velocity vector $\dot{\mathbf{q}}_i$. The omnidirectional wheel Jacobian matrix $\mathbf{J}_i$ is constructed from the transformation matrix ${}^R\mathbf{\Pi}_{C_i}$, the wheel and roller radii R_i and r_i, and the roller angle η_i [1-2]:

$$ \mathbf{J}_i = \begin{pmatrix} -R_i \sin {}^R\theta_{C_i} & r_i \sin({}^R\theta_{C_i} + \eta_i) & {}^Rd_{C_iy} \\ R_i \cos {}^R\theta_{C_i} & -r_i \cos({}^R\theta_{C_i} + \eta_i) & -{}^Rd_{C_ix} \\ 0 & 0 & 1 \end{pmatrix} \; . \qquad (4)$$

Since the determinant of $\mathbf{J}_i$ is $(-r_iR_i \sin \eta_i)$, the omnidirectional wheel Jacobian matrix in (4) is non-singular when $\eta_i \neq 0$ *or* $180°$; i.e., when the rollers are not aligned with the wheel orientation. Since the roller angles on Uranus are $\pm 45°$, the rank of each wheel Jacobian matrix is three, indicating that each wheel has three DOFs (as stated in Section 2).

To simplify fabrication, all wheels are identical. Consequently, the radii of all wheels and rollers are equal (i.e., $R_1 = R_2 = R_3 = R_4 = R$, and $r_1 = r_2 = r_3 = r_4 = r$). The magnitude of the roller angles on all four wheels are also equal; however, wheels 1 and 3 are mounted so that their roller angles are reversed from the roller angles of wheels 2 and 4 (i.e., $\eta_1 = \eta_3 = -45°$ and $\eta_2 = \eta_4 = 45°$). From the Jacobian matrix for omnidirectional wheels in (4), we write the equation-of-motion for each wheel:

$$ \textit{Wheel } 1: \quad {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_1 \dot{\mathbf{q}}_1 \qquad (5)$$

$$ \begin{pmatrix} {}^{\overline{R}}v_{Rx} \\ {}^{\overline{R}}v_{Ry} \\ {}^{\overline{R}}\omega_R \end{pmatrix} = \begin{pmatrix} 0 & -r\sqrt{2}/2 & l_b \\ R & -r\sqrt{2}/2 & -l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_{w_1x} \\ \omega_{w_1r} \\ \omega_{w_1z} \end{pmatrix}$$

$$ \textit{Wheel } 2: \quad {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_2 \dot{\mathbf{q}}_2 \qquad (6)$$

$$ \begin{pmatrix} {}^{\overline{R}}v_{Rx} \\ {}^{\overline{R}}v_{Ry} \\ {}^{\overline{R}}\omega_R \end{pmatrix} = \begin{pmatrix} 0 & r\sqrt{2}/2 & l_b \\ R & -r\sqrt{2}/2 & l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_{w_2x} \\ \omega_{w_2r} \\ \omega_{w_2z} \end{pmatrix}$$

$$Wheel\ 3: \ {}^{R}\dot{\mathbf{p}}_R = \mathbf{J}_3\dot{\mathbf{q}}_3 \qquad (7)$$

$$\begin{pmatrix} {}^{\overline{R}}v_{Rx} \\ {}^{\overline{R}}v_{Ry} \\ {}^{\overline{R}}\omega_R \end{pmatrix} = \begin{pmatrix} 0 & -r\sqrt{2}/2 & -l_b \\ R & -r\sqrt{2}/2 & l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_{w_3x} \\ \omega_{w_3r} \\ \omega_{w_3z} \end{pmatrix}$$

$$Wheel\ 4: \ {}^{R}\dot{\mathbf{p}}_R = \mathbf{J}_4\dot{\mathbf{q}}_4 \qquad (8)$$

$$\begin{pmatrix} {}^{\overline{R}}v_{Rx} \\ {}^{\overline{R}}v_{Ry} \\ {}^{\overline{R}}\omega_R \end{pmatrix} = \begin{pmatrix} 0 & r\sqrt{2}/2 & -l_b \\ R & -r\sqrt{2}/2 & -l_a \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_{w_4x} \\ \omega_{w_4r} \\ \omega_{w_4z} \end{pmatrix} .$$

3.4. Composite Robot Equation

The motion of Uranus results from the simultaneous motions of the wheels. To model the motion, we combine the equations-of-motion of the four wheels in (5)-(8) to form the *composite robot equation-of-motion*:

$$\begin{pmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{pmatrix} {}^{\overline{R}}\dot{\mathbf{p}}_R = \begin{pmatrix} \mathbf{J}_1 & 0 & \dots & 0 \\ 0 & \mathbf{J}_2 & \ddots & \vdots \\ \vdots & \ddots & \mathbf{J}_3 & 0 \\ 0 & \dots & 0 & \mathbf{J}_4 \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \\ \dot{\mathbf{q}}_4 \end{pmatrix} \qquad (9)$$

$$\text{or} \qquad \mathbf{A}_o {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{B}_o\dot{\mathbf{q}} \qquad (10)$$

in which $\mathbf{I}$ is a (3×3) identity matrix, $\mathbf{A}_0$ is a (12×3) matrix, $\mathbf{B}_0$ is a (12×12) block diagonal matrix, and $\dot{\mathbf{q}}$ is the composite wheel velocity vector. The 12 wheel equations in (10) must be solved simultaneously to characterize the WMR motion.

4. Kinematic Solutions

4.1. Actuated Inverse Solution

We compute the actuated inverse solution by calculating the actuated wheel velocities in (10) from the robot velocities. Because of the closed-link chains in WMRs, we need not actuate all of the wheel variables θ_{w_ix}, θ_{w_ir}, and θ_{w_iz}, for $i = 1,2,3,4$. To separate the actuated "a" and unactuated "u" wheel variables, we partition the wheel equation in (3) into two components:

$${}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_{ia}\dot{\mathbf{q}}_{ia} + \mathbf{J}_{iu}\dot{\mathbf{q}}_{iu} \qquad for\ i = 1,2,3,4\ . \qquad (11)$$

We combine the four partitioned wheel equations in (11) to rewrite the composite robot equation in (9) and (10) as

$$(\mathbf{I} \quad \mathbf{I} \quad \mathbf{I} \quad \mathbf{I})^T \ {}^{\overline{R}}\dot{\mathbf{p}}_R$$

$$= \begin{pmatrix} \mathbf{J}_{1a} & 0 & \dots & 0 & \mathbf{J}_{1u} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2a} & \ddots & \vdots & 0 & \mathbf{J}_{2u} & \ddots & \vdots \\ \vdots & \ddots & \mathbf{J}_{3a} & 0 & \vdots & \ddots & \mathbf{J}_{3u} & 0 \\ 0 & \dots & 0 & \mathbf{J}_{4a} & 0 & \dots & 0 & \mathbf{J}_{4u} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1a} \\ \dot{\mathbf{q}}_{2a} \\ \dot{\mathbf{q}}_{3a} \\ \dot{\mathbf{q}}_{4a} \\ \dot{\mathbf{q}}_{1u} \\ \dot{\mathbf{q}}_{2u} \\ \dot{\mathbf{q}}_{3u} \\ \dot{\mathbf{q}}_{4u} \end{pmatrix} \qquad (12)$$

$$\text{or} \qquad \mathbf{A}_0 \ {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{B}_{0p} \begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_u \end{pmatrix} . \qquad (13)$$

In the absence of wheel slippage, the wheel velocities are coupled and hence the problem of calculating the wheel velocities from the robot velocities is *overdetermined*[1-2]; i.e., there are more independent equations than unknowns (wheel velocities). We compute the least-squares solution

$$\begin{pmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_u \end{pmatrix} = (\mathbf{B}_{0p}^T\mathbf{B}_{0p})^{-1} \ \mathbf{B}_{0p}^T \ \mathbf{A}_0 \ {}^{\overline{R}}\dot{\mathbf{p}}_R \qquad (14)$$

to obtain the actuated wheel velocities[1-2]:

$$\dot{\mathbf{q}}_a = \begin{pmatrix} [\mathbf{J}_{1a}^T\Delta(\mathbf{J}_{1u})\mathbf{J}_{1a}]^{-1}\mathbf{J}_{1a}^T\Delta(\mathbf{J}_{1u}) \\ [\mathbf{J}_{2a}^T\Delta(\mathbf{J}_{2u})\mathbf{J}_{2a}]^{-1}\mathbf{J}_{2a}^T\Delta(\mathbf{J}_{2u}) \\ [\mathbf{J}_{3a}^T\Delta(\mathbf{J}_{3u})\mathbf{J}_{3a}]^{-1}\mathbf{J}_{3a}^T\Delta(\mathbf{J}_{3u}) \\ [\mathbf{J}_{4a}^T\Delta(\mathbf{J}_{4u})\mathbf{J}_{4a}]^{-1}\mathbf{J}_{4a}^T\Delta(\mathbf{J}_{4u}) \end{pmatrix} {}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_a \ {}^{\overline{R}}\dot{\mathbf{p}}_R \qquad (15)$$

in which we define the *Delta* matrix function $\mathbf{\Delta}(\bullet)$ as:

$$\mathbf{\Delta}(\mathbf{U}) = \begin{cases} -\mathbf{I} & for \ \ \mathbf{U} = null \\ \mathbf{U}(\mathbf{U}^T \ \mathbf{U})^{-1}\mathbf{U}^T - \mathbf{I} & Otherwise \end{cases} \qquad (16)$$

where the argument $\mathbf{U}$ is a $(c \times d)$ matrix of rank d. In (15), $\mathbf{J}_a$ is the [number of actuators (four for Uranus) $\times$ 3] *actuated* wheel Jacobian matrix.

For Uranus, we identify the actuated wheel velocities as ω_{w_ix}, for $i = 1,2,3,4$, and the unactuated wheel velocities as ω_{w_ir} and ω_{w_iz}, for $i = 1,2,3,4$, and apply the actuated inverse solution in (15) to obtain:

$$\begin{pmatrix} \omega_{w_1x} \\ \omega_{w_2x} \\ \omega_{w_3x} \\ \omega_{w_4x} \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -1 & 1 & (l_a + l_b) \\ 1 & 1 & -(l_a + l_b) \\ -1 & 1 & -(l_a + l_b) \\ 1 & 1 & (l_a + l_b) \end{pmatrix} \begin{pmatrix} {}^{\overline{R}}v_{Rx} \\ {}^{\overline{R}}v_{Ry} \\ {}^{\overline{R}}\omega_R \end{pmatrix} . \qquad (17)$$

Since all of the wheel Jacobian matrices for Uranus are nonsingular, we compute the actuated inverse solution in (17) by solving wheel equations (5)-(8) independently for each of the actuated wheel velocities. This alternate, straightforward approach reduces the computational requirements of our least-squares solution in (11)-(17) by a factor of thirteen.

4.2. Sensed Forward Solution

The sensed forward solution computes the robot velocity vector ${}^{\overline{R}}\dot{\mathbf{p}}_R$ from the sensed wheel positions and velocities $\mathbf{q}_s$ and $\dot{\mathbf{q}}_s$. Development of the sensed forward solution parallels our development of the actuated inverse solution in Section 4.1. The first step is to separate the sensed "s" and not-sensed "n" wheel velocities and rewrite the wheel equations-of-motion in (3) as:

$${}^{\overline{R}}\dot{\mathbf{p}}_R = \mathbf{J}_{is}\dot{\mathbf{q}}_{is} + \mathbf{J}_{in}\dot{\mathbf{q}}_{in} \qquad for\ i = 1,2,3,4\ . \qquad (18)$$

We combine the four wheel equations in (18) with the unknown robot and wheel velocities on the left-hand side:

$$\begin{pmatrix} \mathbf{I} & -\mathbf{J}_{1n} & 0 & \dots & 0 \\ \mathbf{I} & 0 & -\mathbf{J}_{2n} & \ddots & \vdots \\ \mathbf{I} & \vdots & \ddots & -\mathbf{J}_{3n} & 0 \\ \mathbf{I} & 0 & \dots & 0 & -\mathbf{J}_{4n} \end{pmatrix} \begin{pmatrix} {}^{\overline{R}}\dot{\mathbf{p}}_R \\ \dot{\mathbf{q}}_{1n} \\ \dot{\mathbf{q}}_{2n} \\ \dot{\mathbf{q}}_{3n} \\ \dot{\mathbf{q}}_{4n} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{J}_{1s} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2s} & \ddots & \vdots \\ \vdots & \ddots & \mathbf{J}_{3s} & 0 \\ 0 & \dots & 0 & \mathbf{J}_{4s} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_{1s} \\ \dot{\mathbf{q}}_{2s} \\ \dot{\mathbf{q}}_{3s} \\ \dot{\mathbf{q}}_{4s} \end{pmatrix} \qquad (19)$$

$$\text{or} \qquad \mathbf{A}_n \begin{pmatrix} {}^{\overline{R}}\dot{\mathbf{p}}_R \\ \dot{\mathbf{q}}_n \end{pmatrix} = \mathbf{B}_s\dot{\mathbf{q}}_s \ . \qquad (20)$$

Since there are more independent equations (three for each wheel) than unknowns (two not-sensed wheel velocities for each wheel plus three robot velocities), the problem of computing the unknown wheel and robot velocities from the sensed wheel velocities is overdetermined. We thus apply the least-squares solution[1-2]:

$$\begin{pmatrix} {}^{\bar{R}}\dot{\mathbf{p}}_R \\ \dot{\mathbf{q}}_n \end{pmatrix} = \left(\mathbf{A}_n^T \mathbf{A}_n\right)^{-1} \mathbf{A}_n^T \mathbf{B}_s \dot{\mathbf{q}}_s \, , \tag{21}$$

to calculate the robot velocities[1-2]:

$${}^{\bar{R}}\dot{\mathbf{p}}_R = \mathbf{\Psi}^{-1} \, \mathbf{S} \, \dot{\mathbf{q}}_s \tag{22}$$

where $\mathbf{\Psi} = [\Delta(\mathbf{J}_{1n}) + \Delta(\mathbf{J}_{2n}) + \Delta(\mathbf{J}_{3n}) + \Delta(\mathbf{J}_{4n})]$ is a (3×3) matrix and

$$\mathbf{S} = [\Delta(\mathbf{J}_{1n})\mathbf{J}_{1s} \quad \Delta(\mathbf{J}_{2n})\mathbf{J}_{2s} \quad \Delta(\mathbf{J}_{3n})\mathbf{J}_{3s} \quad \Delta(\mathbf{J}_{4n})\mathbf{J}_{4s}] \, .$$

Hence, solving the system of linear algebraic equations in (22) is *not* a computational burden.

For Uranus, we identify the sensed wheel velocities as $\omega_{w_i x}$, for $i = 1, 2, 3, 4$, and the not-sensed wheel velocities as $\omega_{w_i r}$ and $\omega_{w_i z}$, for $i = 1, 2, 3, 4$, and apply the least-squares sensed forward solution in (22) to obtain:

$$\begin{pmatrix} {}^{\bar{R}}v_{Rx} \\ {}^{\bar{R}}v_{Ry} \\ {}^{\bar{R}}\omega_R \end{pmatrix} = \frac{R}{4\, l_{ab}} \begin{pmatrix} -l_{ab} & l_{ab} & -l_{ab} & l_{ab} \\ l_{ab} & l_{ab} & l_{ab} & l_{ab} \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \omega_{w_1 x} \\ \omega_{w_2 x} \\ \omega_{w_3 x} \\ \omega_{w_4 x} \end{pmatrix} , \tag{23}$$

where $l_{ab} = l_a + l_b$. The least-squares forward solution in (23) need not produce a zero error because of sensor noise, wheel slippage, and non-planar surfaces. In the presense of these error sources, we cannot calculate the robot velocity exactly. Our least-squares solution provides an optimal solution by minimizing the sum of the squared errors of the velocity components.

5. Applications

5.1. Dead Reckoning

Dead reckoning[13] is the real-time calculation of the WMR position from wheel sensor measurements. To determine the robot position in real-time, we integrate the robot velocity over each sampling period. The integration begins when the robot is at rest (i.e., ${}^{F}\dot{\mathbf{p}}_R(0) = \mathbf{0}$). The initial robot position ${}^{F}\mathbf{p}_R(0)$ is either specified or determined by computer vision or sonar ranging. We assume that the robot motion is adequately modeled by piecewise constant accelerations since the robot is being actuated by constant force/torque generators in each control sampling period (the control sampling period matches the dead reckoning sampling period). The robot velocity ${}^{\bar{R}}\dot{\mathbf{p}}_R$ *in the* sampling period from time $t = (n-1)T$ to time $t = nT$ is

$$\begin{aligned} {}^{\bar{R}}\dot{\mathbf{p}}_R(t) &= {}^{\bar{R}}\dot{\mathbf{p}}_R[(n-1)T] \\ &+ \frac{{}^{\bar{R}}\dot{\mathbf{p}}_R(nT) - {}^{\bar{R}}\dot{\mathbf{p}}_R[(n-1)T]}{T}(t - [(n-1)T]), \end{aligned} \tag{24}$$

where the robot velocity ${}^{\bar{R}}\dot{\mathbf{p}}_R(nT)$ *at each sampling instant* is calculated according to the sensed forward solution in (23). We apply the orthogonal velocity transformation matrix $\mathbf{V}$ to transform the robot velocity to the floor coordinate system:

$$\begin{aligned} {}^{F}\dot{\mathbf{p}}_R &= \mathbf{V} \, {}^{\bar{R}}\dot{\mathbf{p}}_R \\ &= \begin{pmatrix} \cos {}^{F}\theta_R & -\sin {}^{F}\theta_R & 0 \\ \sin {}^{F}\theta_R & \cos {}^{F}\theta_R & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^{\bar{R}}\dot{\mathbf{p}}_R \, . \end{aligned} \tag{25}$$

We calculate the robot position at the current sampling instant $t = nT$ by integrating the velocity over the sampling period and adding the result to the robot position at the previous sampling instant $t = (n-1)T$:

$${}^{F}\mathbf{p}_R(nT) = {}^{F}\mathbf{p}_R[(n-1)T] + \int_{(n-1)T}^{nT} {}^{F}\dot{\mathbf{p}}_R(t)dt \, . \tag{26}$$

By subtituting (24) and (25) into (26), we express the current robot position in terms of the position at the previous sampling instant and the robot velocity at the current and previous sampling instants:

$$\begin{aligned} {}^{F}\mathbf{p}_R(nT) &= {}^{F}\mathbf{p}_R[(n-1)T] \\ &+ \frac{T}{2}\, \mathbf{V}[(n-1)T] \left\{ {}^{\bar{R}}\dot{\mathbf{p}}_R[(n-1)T] + {}^{\bar{R}}\dot{\mathbf{p}}_R(nT) \right\} . \end{aligned} \tag{27}$$

Upon substituting the sensed forward solution in (23) into (27), expanding, and simplifying, we obtain algorithms for the *direct* calculation of the current orientation ${}^{F}\theta_R(nT)$, and the *recursive* calculation of the current translations ${}^{F}d_{Rx}(nT)$ and ${}^{F}d_{Ry}(nT)$ of Uranus:

$${}^{F}\theta_R(nT) = \frac{R}{4(l_a + l_b)}\, \theta(nT) + {}^{F}\theta_R(0) - \frac{R}{4(l_a + l_b)}\, \theta(0) \tag{28}$$

where $\theta(\bullet) = [\theta_{w_1 x}(\bullet) - \theta_{w_2 x}(\bullet) - \theta_{w_3 x}(\bullet) + \theta_{w_4 x}(\bullet)]$, and

$$\begin{aligned} \begin{pmatrix} {}^{F}d_{Rx}(nT) \\ {}^{F}d_{Ry}(nT) \end{pmatrix} &= \begin{pmatrix} {}^{F}d_{Rx}[(n-1)T] \\ {}^{F}d_{Ry}[(n-1)T] \end{pmatrix} \\ &+ \frac{TR}{8} \begin{pmatrix} -a & b & -a & b \\ b & a & b & a \end{pmatrix} [\dot{\mathbf{q}}_s[(n-1)T] + \dot{\mathbf{q}}_s(nT)] \end{aligned} \tag{29}$$

where $a = [\cos {}^{F}\theta_R(nT) + \sin {}^{F}\theta_R(nT)]$ and $b = [\cos {}^{F}\theta_R(nT) - \sin {}^{F}\theta_R(nT)]$. The direct calculation of the robot orientation indicates that (28) is a *holonomic* constraint[14]. Since the robot translations cannot be calculated directly, (29) are non-holonomic constraints. Errors in the recursively calculated robot translations due to finite precision, sensor noise, wheel slip, and non-planar surfaces will accumulate; whereas, the direct calculation of the holonomic orientation constraint in (28) will not accumulate these errors.

5.2. Wheel Slip Detection

Since the sensed forward solution for Uranus in (23) is overdetermined, the kinematic equations-of-motion are inconsistent in the presence of wheel slip. The error in the least-squares forward solution is then non-zero. We thus detect wheel slip by calculating the error in the least-squares forward solution. If the error in the forward solution exceeds a threshold, we conclude that the forward solution, and hence the dead reckoned robot position, are not sufficiently accurate for feedback control calculations. In the improbable case that all four wheels slip simultaneously in such a manner that the equations-of-motion remain consistent, our approach will fail to detect wheel slip.

In practice, error sources such as finite precision, sensor noise and non-planar surfaces also render the kinematic equations-of-motion inconsistent. We expect that the least-squares error induced by these sources will be small in comparison with the error caused by wheel slippage. We thus compare the least-squares error with an error threshold. If the least-squares error in the forward solution exceeds the threshold, we conclude that wheel slip has occurred. When we detect that wheel slip has occurred, we resort to absolute methods (e.g., computer vision, ultrasonic ranging sensors, and laser range finders) to determine the robot position. Since current absolute locating methods are computationally slow relative to the robot motion, the WMR should halt motion until the dead reckoning algorithm is updated by an absolute locating method.

To calculate the error in the least-squares forward solution, we relate the robot velocity vector (computed by the sensed forward solution) to the sensed wheel velocities by eliminating the not-sensed wheel velocities from the composite robot equation in (10). We express the not-sensed wheel velocities as linear combinations of the robot velocities by applying the actuated inverse solution in (15) with the not-sensed ("n" subscripts) and sensed ("s" subscripts) wheel velocities playing the roles of the actuated ("a" subscripts) and unactuated ("u" subscripts) wheel velocities, respectively:

$$\dot{\mathbf{q}}_n = \begin{pmatrix} [\mathbf{J}_{1n}^T\Delta(\mathbf{J}_{1s})\mathbf{J}_{1n}]^{-1}\mathbf{J}_{1n}^T\Delta(\mathbf{J}_{1s}) \\ [\mathbf{J}_{2n}^T\Delta(\mathbf{J}_{2s})\mathbf{J}_{2n}]^{-1}\mathbf{J}_{2n}^T\Delta(\mathbf{J}_{2s}) \\ [\mathbf{J}_{3n}^T\Delta(\mathbf{J}_{3s})\mathbf{J}_{3n}]^{-1}\mathbf{J}_{3n}^T\Delta(\mathbf{J}_{3s}) \\ [\mathbf{J}_{4n}^T\Delta(\mathbf{J}_{4n})\mathbf{J}_{4n}]^{-1}\mathbf{J}_{4n}^T\Delta(\mathbf{J}_{4n}) \end{pmatrix} {}^{\bar{R}}\dot{\mathbf{p}}_R \ . \tag{30}$$

We partition the sensed and not-sensed wheel velocities in the composite robot equation in (9) and substitute (30) for the not-sensed wheel velocities to obtain the robot *sensing* equation:

$$\begin{pmatrix} \mathbf{I} - \mathbf{J}_{1n}[\mathbf{J}_{1n}^T\Delta(\mathbf{J}_{1s})\mathbf{J}_{1n}]^{-1}\mathbf{J}_{1n}^T\Delta(\mathbf{J}_{1s}) \\ \mathbf{I} - \mathbf{J}_{2n}[\mathbf{J}_{2n}^T\Delta(\mathbf{J}_{2s})\mathbf{J}_{2n}]^{-1}\mathbf{J}_{2n}^T\Delta(\mathbf{J}_{2s}) \\ \mathbf{I} - \mathbf{J}_{3n}[\mathbf{J}_{3n}^T\Delta(\mathbf{J}_{3s})\mathbf{J}_{3n}]^{-1}\mathbf{J}_{3n}^T\Delta(\mathbf{J}_{3s}) \\ \mathbf{I} - \mathbf{J}_{4n}[\mathbf{J}_{4n}^T\Delta(\mathbf{J}_{4n})\mathbf{J}_{4n}]^{-1}\mathbf{J}_{4n}^T\Delta(\mathbf{J}_{4n}) \end{pmatrix} {}^{\bar{R}}\dot{\mathbf{p}}_R$$

$$= \begin{pmatrix} \mathbf{J}_{1s} & 0 & \dots & 0 \\ 0 & \mathbf{J}_{2s} & \ddots & 0 \\ \vdots & \ddots & \mathbf{J}_{3s} & 0 \\ 0 & \dots & 0 & \mathbf{J}_{4s} \end{pmatrix} \dot{\mathbf{q}}_s \ , \tag{31}$$

$$\text{or} \qquad \mathbf{A}_s {}^{\bar{R}}\dot{\mathbf{p}}_R = \mathbf{B}_s \dot{\mathbf{q}}_s \ . \tag{32}$$

Calculation of the sensed forward solution in (22) is the first step in determining the least-squares error. We substitute the computed robot velocity vector ${}^{\bar{R}}\dot{\mathbf{p}}_R$ for the actual robot velocity vector in (32). The least-squares error vector $\mathbf{e}$ is then calculated by subtracting the right-hand side of (32) from the left-hand side:

$$\mathbf{e} \ = \ \mathbf{A}_s \, {}^{\bar{R}}\dot{\mathbf{p}}_R \ - \ \mathbf{B}_s \, \dot{\mathbf{q}}_s \ . \tag{33}$$

We compare the two-norm of the least-squares error $\sqrt{\mathbf{e}^T\mathbf{e}}$ with the scalar threshold e_t. If the norm exceeds the threshold, we conclude that wheel slip has occurred. For Uranus, this algorithm is:

$$If \qquad \frac{R}{2}(\omega_{w_1x} + \omega_{w_2x} - \omega_{w_3x} - \omega_{w_4x}) \ > \ e_t \ ,$$
$$\textit{wheel slip has occurred} \ . \tag{34}$$

5.3. Kinematic Feedback Control

The documented WMR control systems are kinematically based[11-12]; i.e., they do not incorporate a dynamic model of the robot motion. A reference robot trajectory is provided by a trajectory planner and the task of the control system is to produce signals for the wheel actuators so that the WMR tracks the reference trajectory.

A kinematics-based control system for Uranus is diagrammed in Figure 4. Directed arrows indicate the flow of information. The number of scalar variables represented by each arrow is indicated within the flow lines. At time nT, we sample the wheel positions $\mathbf{q}_s(nT)$ and velocities $\dot{\mathbf{q}}_s(nT)$ from the wheel shaft encoders and interface logic, and receive the desired robot position vector ${}^F\mathbf{p}_d(nT)$ from an offboard trajectory planning processor. We apply the dead reckoning algorithm in (28)-(29) to compute the robot position ${}^F\mathbf{p}_R(nT)$. We compare the desired robot position with the actual robot position to calculate the robot position error:

$${}^F\mathbf{e}_R(nT) \ = \ {}^F\mathbf{p}_d(nT) \ - \ {}^F\mathbf{p}_R(nT). \tag{35}$$

The position error is multiplied by the (3×3) diagonal feedforward gain vector $\mathbf{K}$ and is then transformed to the $\bar{r}$obot coordinate frame by applying the inverse motion matrix $\mathbf{V}^{-1}(nT) = \mathbf{V}^T(nT)$ in (25). Under the assumption that the robot tracking error remains small, we treat the robot position error ${}^{\bar{R}}\mathbf{e}_R$ as the differential displacement ${}^{\bar{R}}\delta\mathbf{p}_R$. We apply the actuated inverse solution in (15) to transform this robot differential displacement (as velocities are transformed) into actuator displacements $\delta\mathbf{q}_a$:

$$\delta\mathbf{q}_a = \mathbf{J}_a \ {}^{\bar{R}}\delta\mathbf{p}_R \ . \tag{36}$$

The resulting actuator velocities are then communicated to the pulse-width modulators.

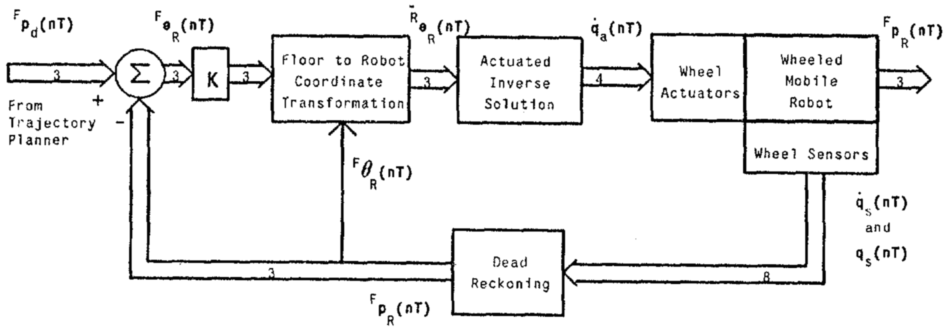

Figure 4: Kinematics-Based WMR Control System

The actuator velocities are calculated by matrix-vector multiplication according to our kinematic feedback control algorithm as:

$$\dot{\mathbf{q}}_a(nT) = \mathbf{J}_a\, \mathbf{V}^T(nT)\, \mathbf{K}\, {}^F\mathbf{e}_R(nT) \qquad (37)$$

which, for Uranus, is:

$$\begin{pmatrix} \omega_{w_1x}(nT) \\ \omega_{w_2x}(nT) \\ \omega_{w_3x}(nT) \\ \omega_{w_4x}(nT) \end{pmatrix} = \frac{1}{R} \begin{pmatrix} -ak_x & -bk_y & lk_z \\ bk_x & ak_y & -lk_z \\ -ak_x & -bk_y & -lk_z \\ bk_x & ak_y & lk_z \end{pmatrix} {}^F\mathbf{e}_R(nT)\,. \quad (38)$$

The elements k_x, k_y, and k_z of the diagonal feedforward gain matrix $\mathbf{K}$ are adjusted experimentally to provide a fast robot tracking response without excessive overshoot or oscillations about the reference trajectory.

6. Summary

We have illustrated our methodology for the kinematic modeling of WMRs[1-2] through the omnidirectional WMR Uranus. We have solved the kinematic equations-of-motion to calculate the actuated inverse and sensed forward solutions. Finally, we applied our kinematic methodology to dead reckoning, wheel slip detection and feedback control algorithm design. These three applications are the fundamental components of the kinematics-based feedback control algorithm (in Figure 4) for Uranus which is executed every sampling period:

(1) Read the present desired robot position vector from the off-board path-planning processor, and sample the positions and velocities of the wheels.

(2) Evaluate (34) to determine whether wheel slip has occurred. If wheel slippage is detected, halt the robot and request the present position of the robot from a vision, sonar, or laser-rangefinding processor.

(3) Execute the dead reckoning algorithm in (28)-(29) to compute the present robot position.

(4) Calculate the robot position error in (35).

(5) Execute the control algorithm in (38) to compute the actuator velocities.

(6) Communicate the resulting actuator velocities to the wheel motors.

Over the past twenty years, manipulator control systems have progressed from independent joint-space control[15], to kinematics-based Cartesian-space control[16], to dynamics-based Cartesian-space feedback control[17], to robust dynamics-based feedback control[18-19] and adaptive control algorithms[20]. Because we anticipate that future WMR control systems will incorporate dynamic models, we are continuing our research by formulating methodologies for the *dynamic modeling* and *dynamics-based feedback control* of WMRs.

7. References

[1] P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," Technical Report No. CMU-RI-TR-86-12, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, July 1986.

[2] P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," *Journal of Robotic Systems*, Vol. 4, No. 2, Spring 1987.

[3] H. P. Moravec (editor), "Autonomous Mobile Robots Annual Report - 1985," Robotics Institute Technical Report No. CMU-RI-MRL-86-1, Carnegie Mellon University, Pittsburgh, PA, January 1986.

[4] Alvema Rehab, "Wheelon - the New Movement," (advertisement), P.O. Box 17017,S-16117 Bromma, Sweden.

[5] B. E. Ilon, "Wheels for a Course Stable Selfpropelling Vehicle Movable in any Desired Direction on the Ground or Some Other Base," U.S. Patent No. 3,876,255 , 1975.

[6] P. F. Muir and C. P. Neuman, "Pulsewidth Modulation Control of Brushless DC Motors for Robotic Applications," *IEEE Transactions on Industrial Electronics*, Vol. IE-32, No. 3, August 1985, pp. 222-229.

[7] M. G. Bekker, *Introduction to Terrain-Vehicle Systems*, The University of Michigan Press, Ann Arbor, MI, 1969.

[8] P. N. Sheth and J. J. Uicker, Jr., "A Generalized Symbolic Notation for Mechanisms," *Journal of Engineering for Industry*, Series B, Vol. 93, No. 70-Mech-19, 102-112 (1971).

[9] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, Vol. 77, No. 2, 215-221 (1955).

[10] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*, The MIT Press, Cambridge, MA, 1981.

[11] T. Hongo, et al., "An Automatic Guidance System of a Self-Controlled Vehicle: The Command System and the Control Algorithm," IEEE Proceedings of the IECON, San Francisco, CA, November 1985, pp. 18-22.

[12] D. J. Daniel, "Analysis, Design, and Implementation of Microprocessor Control for a Mobile Platform," Master's Project Report, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, August 1984.

[13] *McGraw-Hill Encyclopedia of Science & Technology*, 5th Edition, McGraw-Hill Book Company, New York, Vol. 4, pp. 36-38, 1982.

[14] L. D. Landau and E. M. Lifshitz, *Mechanics*, Third Edition, Pergamon Press, New York, 1976.

[15] R. Paul, et al., "Advanced Industrial Robot Control Systems," Technical Report No. TR-EE79-35, School of Electrical Engineering, Purdue University, West Lafayette, IN, 47907, July 1979.

[16] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Protheses," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, No. 2, June 1969, pp.47-53.

[17] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 3, June 1980, pp. 468-474.

[18] V. D. Tourassis and C. P. Neuman, "Robust Nonlinear Feedback Control for Robotic Manipulators," *IEE Proceedings - D: Control Theory and Applications*, Special Issue on Robotics, Vol. 132, No. 4, July 1985, pp. 134-143.

[19] C. P. Neuman and V. D. Tourassis, "Robust Discrete Nonlinear Feedback Control for Robotic Manipulators," *Journal of Robotic Systems*, Vol. 4, No. 1, 1987.

[20] Nicosia, S. and Tomei, P., "Model Reference Adaptive Control Algorithms for Industrial Robots," *Automatica*, Vol. 20, No. 5, September 1984, pp. 635-644.