Slip-Compensated Path Following for Planetary Exploration Rovers

Daniel M. Helmick, Stergios I. Roumeliotis[†], Yang Cheng, Daniel S. Clouse,
Max Bajracharya, and Larry H. Matthies

Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109,
Email: firstname.lastname@jpl.nasa.gov
[†]Department of Computer Science and Engineering, University of Minnesota,
200 Union St. SE, Minneapolis, MN 55455, Email: stergios@cs.umn.edu

Abstract

A system that enables continuous slip compensation for a Mars rover has been designed, implemented, and field-tested. This system is composed of several components that allow the rover to accurately and continuously follow a designated path, compensate for slippage, and reach intended goals in high-slip environments. These components include: visual odometry, vehicle kinematics, a Kalman filter pose estimator, and a slip-compensated path follower. Visual odometry tracks distinctive scene features in stereo imagery to estimate rover motion between successively acquired stereo image pairs. The kinematics for a rocker-bogie suspension system estimates vehicle motion by measuring wheel rates, and rocker, bogie, and steering angles. The Kalman filter processes measurements from an Inertial Measurement Unit (IMU) and visual odometry. The filter estimate is then compared to the kinematic estimate to determine whether slippage has occurred, taking into account estimate uncertainties. If slippage is detected, the slip vector is calculated by differencing the current Kalman filter estimate from the kinematic estimate. This slip vector is then used to determine the necessary wheel velocities and steering angles to compensate for slip and follow the desired path.

keywords: rover navigation, visual odometry, slip compensation, Kalman filter, rover kinematics.

1 INTRODUCTION

This paper describes the design, implementation, and experimental results of an integrated system for Mars rover navigation in high-slip environments (see Figure 1). The presented algorithms enable the rover to accurately follow a designated path, compensate for slippage, and reach intended goals independent of the terrain over which it traverses (within the mechanical constraints of the mobility system). The proposed system is comprised of several key components that were developed and refined

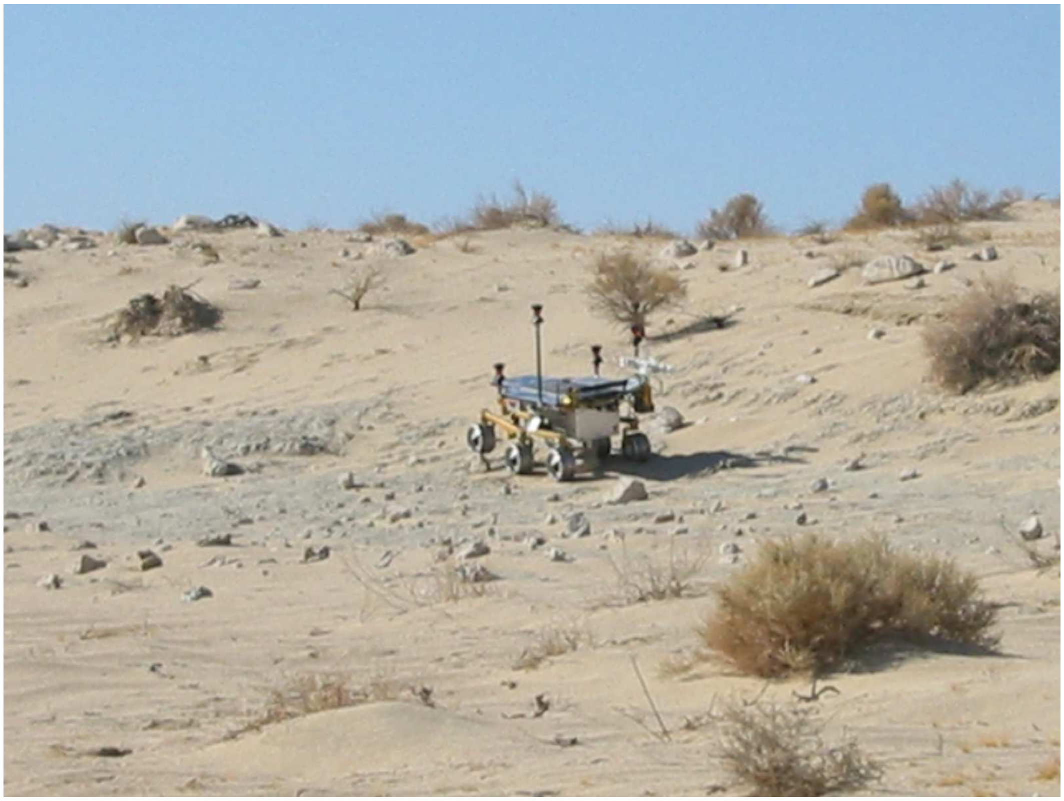

Figure 1: Rocky8 on a Sandy Slope.

for this task and are described in detail below. These components include: visual odometry, full vehicle kinematics, a Kalman filter pose estimator, and a slip-compensated path following algorithm. Figure 2 provides a high-level functional block diagram of the system. *Visual odometry* is an algorithm that relies on stereo imagery to estimate rover motion independent of mechanical terrain properties and is described in Section 2. The full *vehicle kinematics*, described in Section 3, uses position sensor inputs from the joints and wheels of the rocker-bogie mobility system (see Figure 4) to estimate rover motion. The *Kalman filter* combines inertial measurements from the onboard IMU with motion estimates from the visual odometry to estimate rover motion at high sampling rates, and is detailed in Section 4. Since both the IMU measurements and the visual odometry estimates are independent of the vehicle's interaction with the ground, the motion estimates from the Kalman filter can be compared to those computed based on the vehicle kinematics (which are highly dependent upon the vehicle's interaction with its environment) to determine if any statistically significant slippage has occurred. If there is no slippage, the vehicle kinematic motion estimates can contribute to the Kalman filter pose estimates. If, however, slippage is detected, then the kinematic estimate and the Kalman filter estimate are differenced, resulting in a rover *slip vector*. This slip vector is then used by the *path following* algorithm to calculate the rover velocity commands required for following a path while compensating for slip. This algorithm is described in greater detail in Section 6.

The individual components of the system as well as a simplified integrated system have been tested

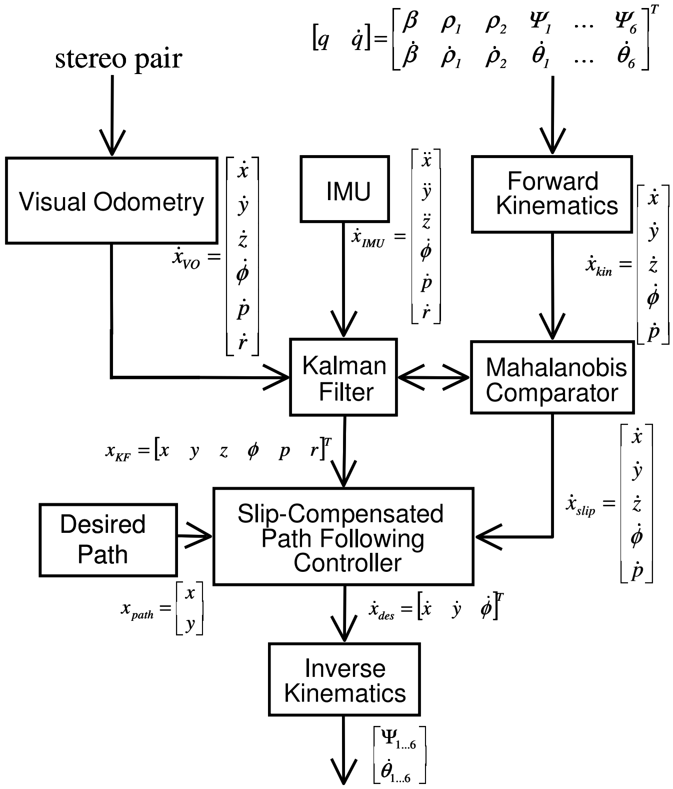
stereo pair
$[q \;\; \dot{q}] = \begin{bmatrix} \beta & \rho_1 & \rho_2 & \Psi_1 & \dots & \Psi_6 \\ \dot{\beta} & \dot{\rho}_1 & \dot{\rho}_2 & \dot{\theta}_1 & \dots & \dot{\theta}_6 \end{bmatrix}^T$
Visual Odometry
$\dot{x}_{VO} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix}$
IMU
$\dot{x}_{IMU} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix}$
Forward Kinematics
$\dot{x}_{kin} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \end{bmatrix}$
Kalman Filter
Mahalanobis Comparator
$x_{KF} = [x \;\; y \;\; z \;\; \phi \;\; p \;\; r]^T$
$\dot{x}_{slip} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \end{bmatrix}$
Desired Path
Slip-Compensated Path Following Controller
$x_{path} = \begin{bmatrix} x \\ y \end{bmatrix}$
$\dot{x}_{des} = [\dot{x} \;\; \dot{y} \;\; \dot{\phi}]^T$
Inverse Kinematics
$\begin{bmatrix} \Psi_{1 \dots 6} \\ \dot{\theta}_{1 \dots 6} \end{bmatrix}$

Figure 2: Block Diagram of the Slip-Compensated Path Following System.

onboard a rover. Several independent tests were performed using Rocky 8 (see Figure 1), a Mars rover research platform. In the first test, visual odometry was tested onboard the rover in the JPL Mars Yard over two 25 meter traverses. Under nominal conditions, wheel odometry accuracy is not better than 10% of the distance traveled and, in higher slip environments, it can be significantly worse. Results from our first tests showed that at least 2.5% accuracy can be achieved by using visual odometry, regardless of the mechanical soil characteristics. The second test was a field test that used the slip compensation system described above, minus the Kalman filter. This test was a traverse of over 50 meters on sandy slopes. The third set of tests was performed on a tiltable platform measuring 5×5 meters. In this last set of tests, the continuous rover-slip compensation algorithm (see Section 6) was employed instead of stopping the rover to estimate slippage, as was the case in previous tests. Results from these three sets of experiments are provided in Section 7. This paper extends the work described in [1, 2]. Related work includes rover trajectory generation [3], rover navigation [4, 5, 6], and path following [7]. To the best of our knowledge this is the first time that an integrated robotic system for slip detection and compensation is being presented and tested in realistic conditions.

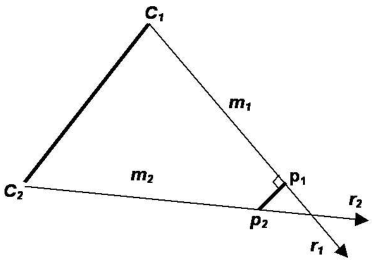

Figure 3: Feature Gap

2 VISUAL ODOMETRY ALGORITHM

Long distance mobile robot navigation on a remote planetary body requires an accurate method for position estimation in an unknown or poorly known environment. Visual odometry, or image-based ego-motion estimation, was originally developed by Matthies [8]. Following this work, a number of variations and modifications have been suggested for improving its robustness and accuracy [9, 10]. The key idea of this method is to determine the change in position and attitude by solving for the maximum likelihood estimate of the transformation between a selection of 3D features extracted from consecutive stereo images. The basic steps of this algorithm are described below. A more detailed description of the ensuing derivations can be found in [2].

2.1 Feature Selection

The first step in the visual odometry algorithm is to select features that can be easily matched between stereo pairs and tracked across a single image step. To achieve this, the Forstner interest operator [12] is applied to the left image of the first stereo pair. The pixels with higher interest values are better features. In order to ensure that the detected visual points are evenly distributed across the image scene, a minimum distance between any two features is enforced. Furthermore, the image scene is divided into grids, with the grid size significantly larger than the minimum distance between features, so as to reduce the volume of data that needs to be sorted. The one feature with the highest interest value in each grid is selected as a viable candidate. Finally, a fixed number of these candidates with the highest interest values which also meet the minimum distance constraint is selected.

2.2 Feature Gap Analysis and Covariance Computation

The 3D positions of the selected features are determined by stereo matching. A template around each feature in the left image is correlated to a location in the right image. Knowing the location of the features in the left and right images, a ray corresponding to the feature can be projected out of each camera. Under perfect conditions, the rays of the same feature from the left and right images should intersect in space. However, due to image noise and matching error, they do not always intersect. The gap (the shortest distance between the two rays) indicates the goodness of the stereo matching. Features with large gaps are eliminated from further processing. Additionally, the error model is a function of the gap. This effect is incorporated in the covariance matrix computation described below.

Assuming the stereo cameras are located at $C_1(X_1, Y_1, Z_1)$ and $C_2(X_2, Y_2, Z_2)$ (see Figure 3), r_1 and r_2 are two unit rays from the same feature in both images. As mentioned before, due to noise, r_1 and r_2 do not always intersect in space. The stereo point is selected to be the midway between the closest points of the two rays.

Assuming the closest points between the two rays are P_1 and P_2 we have:

$$\begin{aligned} P_1 &= C_1 + r_1 m_1 \\ P_2 &= C_2 + r_1 m_2 \end{aligned} \tag{1}$$

where m_1 and m_2 are the lengths of P_1C_1 and P_2C_2, respectively. Since P_1 and P_2 represent the same point in space, we have

$$\begin{aligned} 0 &= (P_2 - P_1) \cdot r_1 = (C_2 - C_1 + r_2 m_2 - r_1 m_1) \cdot r_1 \\ 0 &= (P_2 - P_1) \cdot r_2 = (C_2 - C_1 + r_2 m_2 - r_1 m_1) \cdot r_2 \end{aligned} \tag{2}$$

Solving for m_1 and m_2, yields

$$\begin{aligned} m_1 &= \frac{1}{\delta} \left((r_1 \cdot r_2) r_2 - r_1 \right) \cdot (C_2 - C_1) \\ m_2 &= \frac{1}{\delta} \left(r_2 - (r_1 \cdot r_2) r_1 \right) \cdot (C_2 - C_1) \end{aligned} \tag{3}$$

where $\delta = (r_1 \cdot r_2)^2 - 1$. The coordinates of point P are given by:

$$P = (P_1 + P_2)/2 \tag{4}$$

where P_1 and P_2 are computed by substituting Equations (3) into (1). The covariance for the position of point P is:

$$\Sigma_P = J \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} J^T \tag{5}$$

where J is the Jacobian matrix, i.e., the matrix of partial derivatives of P with respect to the 2D feature locations in the left and right images, and Σ_l and Σ_r are 2×2 matrices whose elements are the curvatures of the biquadratic polynomial along the vertical, horizontal, and diagonal directions, which can be obtained directly from subpixel interpolation. The quality of a 3D feature is a function of its relative location, the gap between the two stereo rays and the sharpness of the correlation peak. This covariance computation fully reflects these three factors. For a more detailed derivation see [2].

2.3 Feature Tracking

After the rover moves some distance, a second pair of stereo images is acquired. The features selected from the previous image are then projected into the second pair using the knowledge of the approximated motion provided by the onboard wheel odometry (forward kinematics). The features are first matched in the new left image by searching an area around the projected feature locations. Stereo matching is then performed on these tracked features on the second pair to determine their new 3D positions. Because the 3D positions of those tracked features are already known from the previous step, the stereo matching search range can be greatly reduced. Features whose initial and final 3D positions differ by too large an amount are filtered out.

2.4 Motion Estimation

Given two sets of corresponding 3D features, the transformation between them is determined using a motion estimation algorithm that takes into account the matching covariance of each feature. This algorithm is decomposed into two sequential steps. Coarse motion is first computed with Schonemann motion estimation, and then a more accurate motion estimate is determined by maximum likelihood motion estimation.

Schonemann motion estimation [13] uses singular value decomposition (SVD) with an orthogonal constraint to estimate the rotation matrix, R, and translation vector, T, that transforms the feature positions in I_1 to those found in I_2. The Schonemann method is simple and fast, however, it is highly unstable when large errors are involved. In order to overcome this problem, a least-median-of-squares approach [14] is adopted. In this method, a subset of features is randomly selected. Then each feature from the previous frame is projected to the current frame, and the distance error between that projection and the position of the corresponding feature in I_2 is computed. The total count of features under a given error tolerance is calculated. This procedure is repeated multiple times. The motion with the largest number of agreeable features is chosen as the best motion.

The best motion estimate determined by employing the above procedure is refined using Maximum Likelihood (ML) motion estimation. The ML estimation algorithm takes into account the 3D feature position differences and the associated error models in order to estimate motion. Let $P_j^{(1)}$ and $P_j^{(2)}$ be the observed feature positions before and after a robot motion. Then we have

$$P_j^{(2)} = RP_j^{(1)} + T + e_j \tag{6}$$

where $R = R(\Theta)$ and T are the rotation and translation of the robot and e_j is the combined errors in the observed positions of the j^{th} features. In this estimation process, the 3-axis rotations (Θ) and translation (T) are directly determined by minimizing the summation in the exponents $\Sigma_j e_j^T W_j e_j$ where $e_j = P_j^{(2)} - RP_j^{(1)} - T$ and W_j is the inverse covariance matrix of e_j. The minimization of this nonlinear problem is implemented by iterative linearization [8]. Two nice properties of ML estimation make this algorithm significantly more accurate compared to the Schonemann method. First, it estimates the 3-axis rotations (Θ) directly so that it eliminates the error caused by rotation matrix estimation (which occurs with least-squares estimation). Second, it incorporates error models for the feature positions in the estimation process (see Equation (5)), which greatly improves the achieved accuracy.

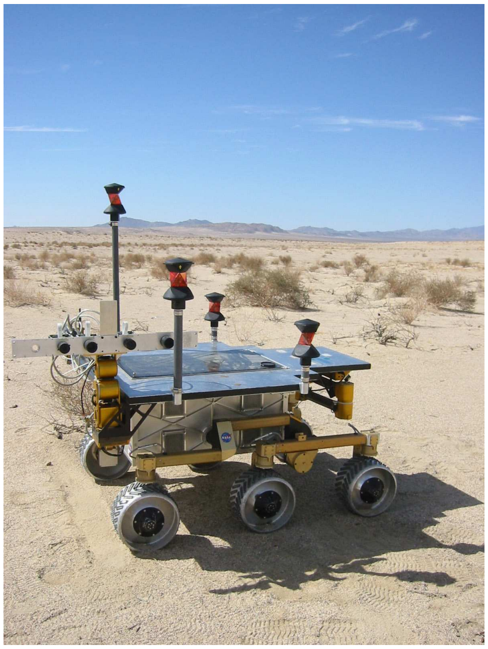

Figure 4: Rocky 8.

3 KINEMATIC ALGORITHMS

The forward and inverse kinematics were determined for the rover (Rocky 8) shown in Figure 2. The forward kinematics of the vehicle are used for estimating rover motion given the wheel rates and rocker, bogie, and steering angles. The inverse kinematics of the vehicle are employed for calculating the necessary wheel velocities and steering angles to create a desired rover motion.

These algorithms are specific to the rocker-bogie configuration with six steerable wheels (see Figure 4). However, the techniques used to derive these algorithms could be applied to any vehicle configuration, though there may be a fewer number of observable degrees of freedom (DOFs) for different configurations. Additionally, these forward kinematic algorithms can be used directly for rovers with a subset of functionality (e.g., a rocker-bogie rover with only four of six wheels capable of steering, such as the Mars Exploration Rovers; or a rover with only four wheels with or without a rocker, etc.) simply by restricting the relevant parameters to be constant.

The motivation for developing the full kinematics of this class of vehicles, rather than adopting the more common planar simplification, is twofold. First, it allows for the observation of 5 DOFs, whereas the planar approximation limits this to 3 DOFs. Second, as terrain becomes rougher, the errors due to the planar assumption grow. These errors can become significantly large (up to 30% of the distance traveled) and affect the slip calculations and, consequently, the slip compensation controller.

The formulation of the forward and inverse kinematics closely follows that of [15, 16], with significant extensions being made for 6-wheel steering. A detailed description of the kinematic derivations can be found in [2, 15, 16, 17].

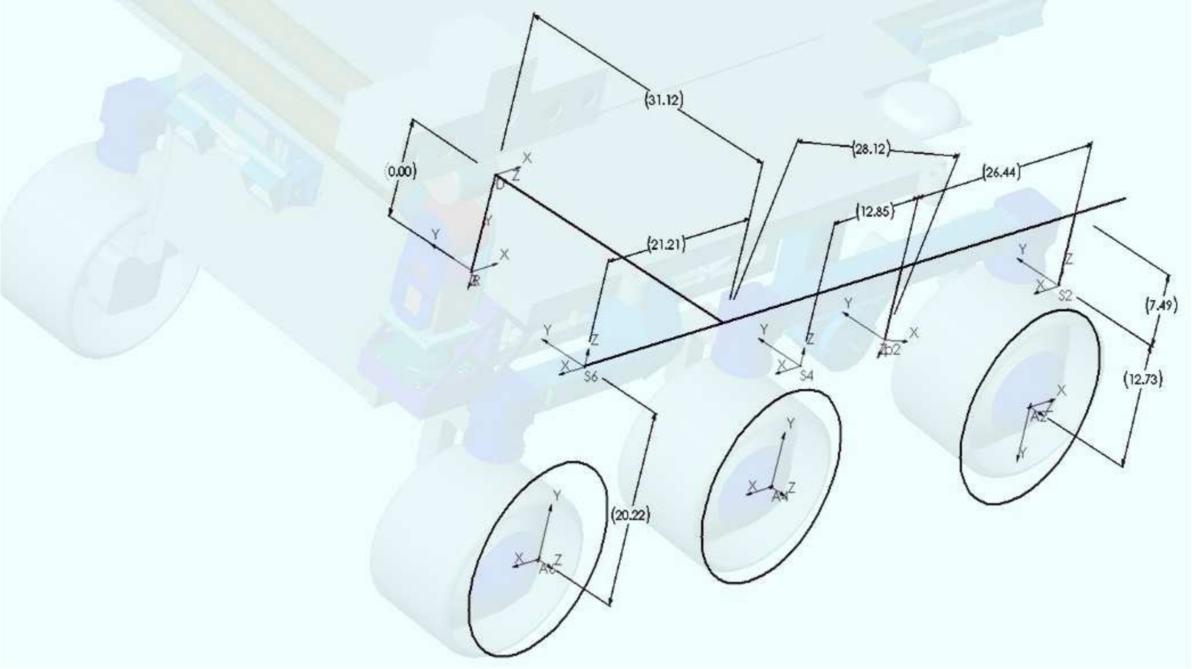

Figure 5: Coordinate Frame Definition for the Right Side of the Rover (all dimensions in cm).

3.1 Rocker-Bogie Configuration

The rocker-bogie configuration is a suspension system that is commonly used for planetary rovers and their prototypes. The configuration analyzed in this work consists of 15 DOFs: 6 steerable/drivable wheels (12 DOFs), a rocker, and two bogies (all of these DOFs are sensed using either encoders or potentiometers).[1] In what follows, we show that under two assumptions the rocker-bogie system allows for the observation of 5 of the 6 DOFs of the rover. These assumptions are: 1) the wheel/terrain contact point is in a constant location relative to the wheel axle, and 2) slip between the wheel and the terrain only occurs about the steering axis (e.g., no side or rolling slip). These slip assumptions, however, are only made for the kinematics algorithm and not for the slip compensation system as a whole. The second assumption is what makes the difference between the Kalman filter motion estimate and the kinematic motion estimate a measurement of vehicle slippage.

3.2 D-H Table Formulation

The Denavit-Hartenberg convention was adopted for defining the frames of each of the 15 DOFs [19]. From the frame definitions (see Figure 5), a unique set of D-H parameters can be derived that completely describes the kinematics of the rover. From these parameters, wheel Jacobians are computed as described in [2].

[1] It is beyond the scope of this paper to describe the benefits of such a mobility system; the interested reader is referred to [18] for a discussion on this topic.

3.3 Forward Kinematics

Once the wheel Jacobians are known, rover motion estimation can be performed using the least-squares formulation

$$\begin{bmatrix} v \\ \dot{\eta} \end{bmatrix} = (A^T A)^{-1} A^T \cdot J_{comp} \dot{q}_{comp} \tag{7}$$

where $v = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{p} & \dot{r} \end{bmatrix}^T$ is the vector of rover linear and rotational velocities, $\dot{\eta}$ is a vector of unobservable wheel turning rates (the sum of the rotational slip rate and the steering rate), A is the matrix of the unsensed elements of the Jacobians, J_{comp} is the block-diagonal composite matrix of the wheel Jacobians, and $\dot{q}_{comp}$ is the composite vector of the measured joint kinematic rates. For details on the derivation and simplified solution to this least-squares problem see [15, 17]. Note that it is not necessary to actually perform the inversion of $A^T A$. The matrix equations can be greatly simplified algebraically to make these computations significantly more efficient.

3.4 Inverse Kinematics

As depicted in Figure 2, the inverse kinematics algorithm receives as input the commanded rover motion, and the current kinematic angles and angle-rates, and produces six steering angles and six wheel rates. An interesting feature of the six steerable wheels is the fact that this configuration creates a holonomic rover, under the assumption of instantaneous steering. Consequently all three controllable DOFs of the rover, $[\ \dot{x},\ \dot{y},\ \dot{\phi}\ ]$, are independent, which enables the isolation of several different control loops as described in Section 6.2.

The first step of the inverse kinematics algorithm is to calculate an instantaneous center of rotation, $[\ x_O\ y_O\ ]$, in the rover frame, by employing the following equations

$$\begin{aligned} x_O &= \dot{y}_{cmd} / \dot{\phi}_{cmd} \\ y_O &= \dot{x}_{cmd} / \dot{\phi}_{cmd} \end{aligned} \tag{8}$$

where $\dot{x}_{cmd}$, $\dot{y}_{cmd}$, and $\dot{\phi}_{cmd}$ are the commanded vehicle velocities from the slip-compensated path follower (see Section 6). Because the z-axis of each wheel is the steerable axis, the instantaneous center of rotation needs to be projected from the rover frame $\{R\}$ to the x-y plane of the steering contact coordinate frame $\{M_i\}$, $i = 1 \ldots 6$, by employing the first two rows of the transformation matrix $T_R^{M_i}$ between these frames:

$$\begin{bmatrix} x_O \\ y_O \end{bmatrix}_{M_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot T_R^{M_i} \cdot \begin{bmatrix} x_O \\ y_O \\ 0 \\ 1 \end{bmatrix}_R \tag{9}$$

The steering angle for each wheel is then calculated using

$$\psi_i = \arctan(x_{O,M_i} / y_{O,M_i}) \ , \quad i = 1 \ldots 6 \tag{10}$$

and each wheel rate is determined as

$$\dot{\theta}_i = [J_{ai}^T \Delta(J_{ui}) J_{ai}]^{-1} J_{ai}^T \Delta(J_{ui}) E \cdot v_{cmd} \ , \quad i = 1 \ldots 6 \tag{11}$$

where

$$\Delta(J_{ui}) = J_{ui}(J_{ui}^T J_{ui})^{-1} J_{ui}^T - I \ , \tag{12}$$

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \ , \tag{13}$$

$$v_{cmd} = \left[\begin{array}{ccc} \dot{x}_{cmd} & \dot{y}_{cmd} & \dot{\phi}_{cmd} \end{array} \right]^T \ , \tag{14}$$

and J_{ai} and J_{ui} are the actuated and un-actuated Jacobians defined in [17]. Note that Equation (11) is the actuated inverse solution from Muir and Neumann [17]. As before, these matrix inversions can be algebraically simplified so that each wheel-rate calculation is relatively simple and computationally efficient.

4 KALMAN FILTER

In this section, we present our approach for estimating the position and orientation of the rover using inertial measurements from the IMU and relative pose (position and orientation measurements) from visual odometry and vehicle odometry (forward kinematics). Since our formulation is based on sensor modeling, we use the indirect form of the Extended Kalman Filter (EKF) that estimates the errors in the estimated states instead of the states themselves. The interested reader is referred to [21, 23, 24] for a detailed description of the advantages of the *indirect* KF vs. the *direct* KF. Within this framework, the IMU measurements are integrated in order to propagate the state estimate [20, 22], while the visual odometry and vehicle kinematics (only when no slip has occurred), are employed for updating the state estimate and providing periodic corrections. The equations of the EKF designed for this particular nonlinear system are described in detail in [22].

4.1 System Propagation Model

The state vector of interest in this estimation problem is:

$$x^T = \left[\begin{array}{ccccc} q^T & b_g^T & v^T & b_a^T & p^T \end{array} \right] \tag{15}$$

where q is the 4×1 quaternion of rotation that represents the attitude of the vehicle, v and p are the 3×1 vectors of linear velocity and position of the rover, and b_g and b_a are the 3×1 vectors of biases in the gyroscope and accelerometer signals. The corresponding error state vector is:

$$\Delta x^T = \left[\begin{array}{ccccc} \delta\theta^T & \Delta b_g^T & \Delta v^T & \Delta {b_a}^T & \Delta p^T \end{array} \right] \tag{16}$$

where $\Delta\zeta = \zeta - \hat{\zeta}$ is the algebraic difference (error) between the real value of a state $\zeta \in \{b_g,\ v,\ b_a,\ p\}$ and its estimate $\hat{\zeta}$, and $\delta\theta$ is the 3×1 error vector of the tilt angles determined based on the small angle approximation:

$$q = \delta q \otimes \hat{q}, \quad \delta q^T = \left[\begin{array}{cc} (1/2)\delta\theta^T & 1 \end{array}\right] \tag{17}$$

The continuous-time equation for the error-state propagation is

$$\delta\dot{x}(t) = F_c(t)\delta x(t) + G_c(t)w(t) \tag{18}$$

where $w(t)$ is the white zero-mean continuous time noise process affecting the IMU signals,

$$F_c(t) = \begin{bmatrix} -\lfloor\hat{\omega}\rfloor & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -C^T(\hat{q})\lfloor\hat{a}\rfloor & 0_{3\times3} & 0_{3\times3} & -C^T(\hat{q}) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \quad G_c(t) = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -C^T(\hat{q}) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix},$$

$\lfloor\hat{\omega}\rfloor$ ($\lfloor\hat{a}\rfloor$) is the skew-symmetric matrix of the vector $\hat{\omega} = \omega_m - \hat{b}_g$ ($\hat{a} = a_m - \hat{b}_a$), ω_m (a_m) is the measured rotational velocity (linear acceleration) vector, and $C^T(\hat{q})$ is the rotational matrix that projects vectors from the sensor to the global coordinate frame. By discretizing Equation (18) we obtain:

$$\delta x_{k+1} = F_k(t_{k+1}, t_k)\delta x_k + w_k \tag{19}$$

where $F_k(t_{k+1}, t_k) = e^{\int_{t_k}^{t_{k+1}} F_c(\tau)d\tau}$, and w_k is the equivalent discrete-time white zero-mean noise process with covariance $Q_k = \int_{t_k}^{t_{k+1}} F_k(t_{k+1},\tau)G_c(\tau)Q_cG_c^T(\tau)F_k^T(t_{k+1},\tau)d\tau$, Q_c is a block diagonal matrix with matrix elements $\sigma_i^2 I_{3\times3}, i = 1\ldots4$, and σ_i^2 is the power of the gyroscope, gyroscope-bias, accelerometer, and accelerometer-bias continuous-time noise components, respectively. The interested reader is referred to [22] for the details of the derivation for Equations (18) and (19).

4.2 Relative Pose Measurement Model

The motion measurement provided by the visual odometry and/or the forward kinematics of the vehicle corresponds to a relative pose measurement, i.e., the algebraic difference in position z_p and attitude quaternion z_q. In order to process measurements that relate state estimates at different time steps (e.g., t_k and t_{k+m}), the state vector needs to be augmented with a copy of the vehicle state estimate at t_k. This process enables the filter to explicitly account for the correlations of the vehicle state estimates between the time instants that e.g., two sets of images were recorded [25]. In what follows, we assume that at time t_k the vehicle is at position ${}^Gp(t_k) = p_1$ with quaternion attitude ${}^1_Gq(t_k) = q_1$, and after m steps it has moved to position ${}^Gp(t_{k+m}) = p_2$ with attitude ${}^2_Gq(t_{k+m}) = q_2$. Frames {G}, {1}, and {2} are the inertial frames of reference attached to the vehicle at times t_0, t_k, and t_{k+m} respectively.

The errors in the relative position and attitude (pose) measurements are given by:

$$\begin{aligned}\Delta\tilde{z}_{k+m} &= \begin{bmatrix}\Delta z_p \\ \Delta\tilde{z}_q\end{bmatrix} = \mathcal{X}\begin{bmatrix}\Delta z_p \\ \Delta z_q\end{bmatrix} \\ &= \Gamma\begin{bmatrix} J_{x_1} & J_{x_2}\end{bmatrix}\begin{bmatrix}\Delta x_1 \\ \Delta x_2\end{bmatrix} + \mathcal{X} n_r = H\begin{bmatrix}\Delta x_1 \\ \Delta x_2\end{bmatrix} + \tilde{n}_r \qquad (20)\end{aligned}$$

where Γ is a 6×6 block diagonal matrix with both 3×3 matrix elements equal to the rotational matrix ${}^{1}_{G}C(q)$, and J_{x_1}, J_{x_2} are the measurement Jacobians with respect to the state vector at the two time instants t_k and t_{k+m}, correspondingly. In this last expression, $\mathcal{X}$ is the 6×7 block diagonal matrix with matrix elements the 3×3 identity matrix and the 3×4 Jacobian matrix that projects the algebraic quaternion difference to the tilt angles' error. All these quantities are described in detail in [22]. Both the original measurement noise n_r and the projected $\tilde{n}_r$ are assumed to be a zero-mean white noise Gaussian processes with covariances:

$$R_r = E[n_r n_r^T] = \begin{bmatrix} R_q & R_{pq} \\ R_{pq} & R_q\end{bmatrix}, \; \tilde{R}_r = E[\tilde{n}_r\tilde{n}_r^T] = \mathcal{X}R_r\mathcal{X}^T$$

As is evident from Equation (20), the relative pose measurement error is expressed in terms of the current $\triangle x_2 = \triangle x(t_{k+m})$ and the previous $\triangle x_1 = \triangle x(t_k)$ (error) state of the system. Note that t_k and t_{k+m} are the time instants when, e.g., the two images (encoder readings) processed by the visual (vehicle) odometry algorithm were recorded and thus the relative pose (motion estimate) measurement provided by it corresponds to the time interval [t_k t_{k+m}].

4.3 Augmented-state propagation

If $\triangle x_{k/k}$ is the state estimate at time t_k (when the first image or encoder measurement was recorded) we augment the state vector with a second copy of this estimate:

$$\triangle\breve{x} = \begin{bmatrix}\triangle x_{k/k}^T & \triangle x_{k/k}^T\end{bmatrix}^T$$

Since initially, at time t_k, the two copies of the state estimate are identical, the covariance matrix for the augmented system would be:

$$\breve{P}_{k/k} = \begin{bmatrix} P_{kk} & P_{kk} \\ P_{kk} & P_{kk}\end{bmatrix}$$

where P_{kk} is the covariance matrix for the (error) state of the vehicle at the time t_k. In order to conserve the estimate of the state at t_k, necessary for evaluating the relative pose measurement error at t_{k+m}, only the second copy of the state estimate is propagated during this interval, while the first remains stationary. The propagation equation for the augmented system is:

$$\begin{bmatrix}\Delta x_s \\ \Delta x\end{bmatrix}_{k+1/k} = \begin{bmatrix} I & 0 \\ 0 & F_{k+1}\end{bmatrix}\begin{bmatrix}\Delta x_s \\ \Delta x\end{bmatrix}_{k/k} + \begin{bmatrix}0 \\ I\end{bmatrix} w_k$$

or

$$\Delta\breve{x}_{k+1/k} = \breve{F}_{k+1}\Delta\breve{x}_{k/k} + \breve{G}_{k+1}w_k$$

where Δx_s is the stationary copy of the error state of the vehicle and w_k is the system noise due to errors in the IMU measurements [22]. The covariance of the augmented system is propagated and after m steps it is:

$$\breve{P}_{k+m/k} = \begin{bmatrix} P_{kk} & P_{kk}\mathcal{F}^T \\ \mathcal{F}P_{kk} & P_{k+m/k} \end{bmatrix} \tag{21}$$

where $\mathcal{F} = \prod_{i=1}^{m} F_{k+1}$ and $P_{k+m/k}$ is the propagated covariance of the evolving state at time t_{k+m}.

At this point, we should note that by employing the error models for the propagation and update (see Section 4.2), the state vector can be updated every time a relative pose measurement from the visual odometry or the kinematics algorithm becomes available. The equations for the augmented-state Kalman filter are described in detail in [22, 25].

5 MAHALANOBIS COMPARATOR (SLIP ESTIMATION)

In this section, we present our approach to rover slippage detection. Based on the kinematic equations of the rover and assuming no wheel slippage, the wheel and rocker-bogie joint measurements are processed to produce a relative position and orientation measurement over a certain time (sampling) interval. Before updating the state estimate of the EKF, this kinematics-based displacement measurement needs to be validated by comparing it to the equivalent one estimated by the EKF. If significant wheel slippage has occurred, the residual (i.e., difference between odometry-based and EKF-based displacement) will be significantly larger compared to the case where the rover moves on solid ground without any of the wheels slipping. A statistical measure for assessing the validity of these measurements is the Mahalanobis distance

$$m_d = r_{k,k+m}^T S^{-1} r_{k,k+m} \tag{22}$$

where $r_{k,k+m}$ is the residual and S is the corresponding residual covariance matrix. In the case of this vehicle's odometry measurement, the Mahalanobis distance follows a chi-square distribution with five degrees of freedom. A sufficient test for validating vehicle odometry measurements $z_{k,k+m}$ is to require that these match the expected (estimated by the EKF) measurements $\hat{z}_{k,k+m}$ of the same quantities with a certain level of confidence. By requiring the fit between the expected and actual measurements to be valid with probability, e.g., p=95%, odometric measurements are processed by the EKF only when $m_d \leq t$, with t = 11.07. If this inequality does not hold, these measurements are discarded and wheel slippage is detected. In this case, the residual is provided to the slip compensation algorithm for appropriately modifying the rover steering and driving commands.

The incentive to include the kinematics measurement in the Kalman filter estimator, when it passes the Mahalanobis distance criterion (as shown in Figure 2), is mainly for off-nominal conditions, when visual odometry does not perform at 2.5% accuracy (for example, under rare conditions, when it does

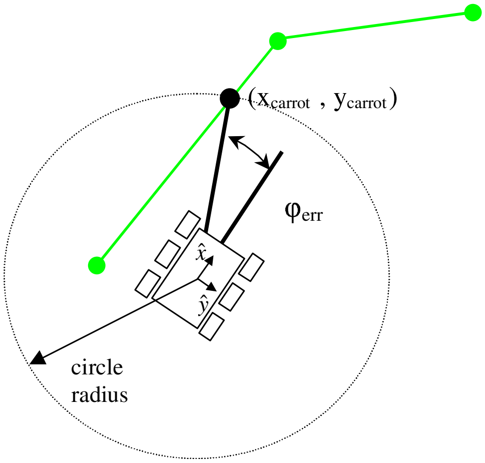

Figure 6: Carrot Heading Calculation.

not converge to a solution). Under nominal conditions for visual odometry, and when the vehicle is slipping, is when the vehicle kinematics are most useful since they allow for the estimation of slippage.

6 SLIP-COMPENSATED PATH FOLLOWER

At the center of Figure 2 is the *slip-compensated path following controller*. This is comprised of two algorithms that are used in close conjunction to achieve the system's end-goal of enabling the traversal of a desired path through high-slip environments. At the highest level, the algorithms receive a 3×1 slip vector $\begin{bmatrix}\dot{x} & \dot{y} & \dot{\phi}\end{bmatrix}_{slip}$, a 3×1 rover pose vector $\begin{bmatrix}x & y & \phi\end{bmatrix}_{pose}$, and a 2×$n$ desired path vector (where n is the number of waypoints) $\begin{bmatrix}x & y\end{bmatrix}_{path}$, and then outputs a 3×1 commanded rover velocity vector $\begin{bmatrix}\dot{x} & \dot{y} & \dot{\phi}\end{bmatrix}_{cmd}$.

6.1 Carrot Heading Algorithm

The *carrot heading* algorithm calculates the desired heading, ϕ_{carrot}, of the vehicle given the desired path and the current rover pose. This algorithm was employed for its robustness to path error [26, 27]. The desired path consists of a set of linear segments between waypoints. The waypoints, however, can be spaced at any distance apart, thus allowing for paths of arbitrary complexity. The algorithm determines the desired heading by calculating the intersection of a circle centered on the rover frame origin with the desired path and computing the direction of that point (Figure 6). The intersection point, $(x_{carrot},\ y_{carrot})$, that is furthest along the path is always selected. The heading error is then computed as:

$$\phi_{err} = \phi_{carrot} - \phi_{pose} \tag{23}$$

A large radius results in a smooth motion of the rover, but tends to filter out small features of the path. A small radius reduces the total path following error. It requires, however, large heading changes of the rover for small path errors, which is extremely inefficient. A circle radius is selected that balances the desire to closely follow the path and the magnitude of the heading changes. Under nominal conditions, the rover path error will always be smaller than the circle radius. If this is not the case, then the radius is grown until an intersection occurs.

6.2 Slip-Compensated Path Following Algorithm

When the Mahalanobis comparator determines that slippage has actually occurred, the calculation of rover slip is made by comparing the output from the Kalman Filter and the output from the forward kinematics. If statistically significant slippage is not detected then the slip vector consists of zeros and the compensation algorithm described below converges to a heading controller.

The slip compensation algorithm consists essentially of two separate control loops. The first control loop, the heading controller, is described by the equation:

$$\dot{\phi}_{cmd} = (K_1 \cdot \phi_{err} + K_2 \cdot \dot{\phi}_{slip})/T_s \tag{24}$$

This loop determines the commanded yaw rate of the vehicle as a combined function of the heading error, ϕ_{err} (as calculated by the carrot heading algorithm), and the yaw slip rate, $\dot{\phi}_{slip}$. It attempts to achieve the optimal heading, determined by the carrot algorithm, even when slipping in the yaw direction. The second loop is described by the equation:

$$\dot{y}_{cmd} = K_3 \cdot \dot{y}_{slip}/T_s \tag{25}$$

This loop calculates the rate of the rover along the y-axis based entirely on the slip in the y direction during the previous sample period. A $\dot{y}_{cmd}$ command results in a crabbing maneuver, where all six wheels have a steering angle offset in the same direction. $\dot{x}_{slip}$ is implicitly compensated for by the rover driving for a longer period of time along the path towards the goal. In Equations (24) and (25), K_1, K_2, and K_3 are tuned controller gains and T_s is the controller sample period. $\dot{x}_{cmd}$ is then determined to be the maximum value allowed that keeps the rover within its operational constraints (i.e., the maximum speed of the drive motors). These rover commands, $[\ \dot{x}\ \dot{y}\ \dot{\phi}\ ]_{cmd}$, are then passed to the inverse kinematics.

6.3 Continuous Motion

With minor changes made to the algorithm it was possible to allow continuous motion of the rover while compensating for slippage. Images are taken, visual odometry is run, the slip vector is estimated, and the compensation command is then computed, all while the rover is moving.

There are many benefits for selecting a continuous motion profile: (i) more efficient motion (both in energy and time); (ii) less slippage due to smaller changes in momentum (this becomes particularly significant as the rover increases in mass and the terrain becomes steeper); and (iii) it enables a higher rate of slip compensation, thus achieving greater accuracy when following a path. It also removes the requirement of many navigation algorithms to use arcs as the fundamental motion of the rover and

Table 1: Rocky 8 Rover Specifications

number of driveable wheels	6
number of steerable wheels	6
wheel diameter (meters)	0.20
wheel width (meters)	0.10
mass (kg)	60
wheelbase (meters)	0.75
track width (meters)	0.62
maximum linear velocity (meters/sec)	0.09
maximum rotational velocity (rad/sec)	0.24
camera resolution (pixels)	640×480
camera field of view (horizontal °× vertical °)	79.5×64.0
camera baseline (meters)	0.084
camera angle from horizontal (°)	45.0
processor type and speed (GHz)	PentiumIII 1.2

permits more complex spline paths to be the basis of path planning. Another advantage is that it supports more optimal spacing between navigation stops (allowing for other requirements, such as IMU bias zeroing, to determine the appropriate spacing between stops).

7 EXPERIMENTAL RESULTS

Three sets of experiments have been performed using Rocky 8 (Figure 4), a Mars rover research platform developed at JPL. This rover has a very similar mobility system to Sojourner, Mars Exploration Rovers (MER), and the current design of the 2009 Mars Science Laboratory (MSL) rover. Rocky 8 utilizes a rocker-bogie suspension (see Section 3.1) which allows for excellent mobility over very rough terrain [18]. Specifications of the Rocky 8 rover are in Table 1.

With the processor onboard the rover, visual odometry runs at approximately 1 Hz. This rate dictates the sample period of the slip-compensated path follower, T_s. Several parameters of the slip-compensated path follower also had to be tuned in the field. The three gains of the control loops K_1, K_2, and K_3 in Equations (24) and (25) were tuned by first increasing K_1 until the heading error was maintained at an acceptably low value without inducing oscillations in the heading of the rover. Then K_2 was increased until the rover acceptably compensated for yaw slippage under the conditions experienced in the field. Finally K_3 was increased until the slippage in the y direction was effectively compensated for without inducing crabbing oscillations. Additionally, a radius of the carrot heading algorithm was determined that balanced the need to closely follow the path with the need to minimize heading changes.

The first set of experiments was performed in the JPL Marsyard, a 20×20 meter space designed as an

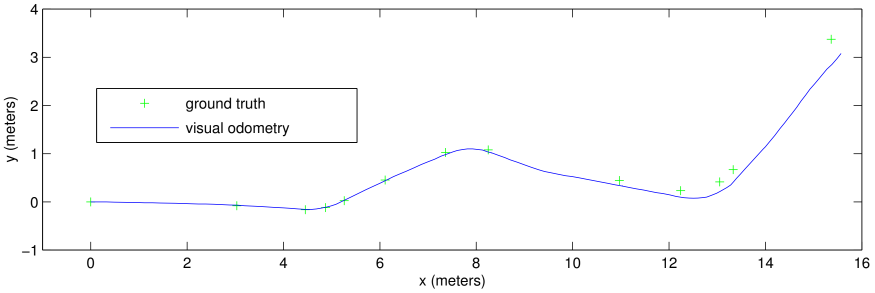

Figure 7: Mars Yard Run 1 Visual Odometry Results.

analogue (in rock size/distribution and soil characteristics) to the Viking Lander sites. The experiment consisted of two consecutive 25-meter runs with visual odometry running onboard. The second set of experiments was performed in the Mojave Desert. The terrain of this area contains slopes up to 25° consisting of loose granular sand (Figure 1). This set of experiments was a test of a simplified integrated slip-compensated path following system. It was simplified in the sense that the Kalman filter and Mahalanobis comparator had not yet been implemented, however, a slip estimate was calculated and compensated for every time the visual odometry provided a new estimate, which was approximately every 20-30 cm. Another simplification, due to limitations of the vehicle, was to assume the rocker and bogie angles were zero. The third set of experiments was performed on JPL's tiltable platform, a 5×5 meter "sandbox" that could be tilted in discrete increments from 0° to 30°. The terrain on the tiltable platform can be characterized as cohesionless beach sand. In the experiments presented here it was set at 10°. This set of experiments consisted of multiple runs at varying approach angles (ranging from straight up the 10° slope to horizontally across the slope), and used the updated continuous motion algorithm described in Section 6.2.

In the first two sets of experiments, ground truth data were collected with a Leica Total Station (LTS), which is a laser-based position measurement system. The LTS was used to measure the absolute position of four prisms mounted on the vehicle (Figure 4) every time the rover was stationary (approximately every 20-30cm). This system gives an accuracy of ±2 mm in position and ±0.2° in attitude. In the field test experiments the waypoints for the rover were also designated using the LTS and a single prism. In the third set of experiments, the LTS could only be used for ground truth at the beginning and end of the runs because the rover did not stop during each test.

7.1 Visual Odometry Results

Visual odometry results are shown from the Marsyard (Figures 7 and 8) and the Mojave Desert experiments (Figure 9). The errors at the end of both Marsyard runs are less than 2.5% of the distance traveled. As shown in Figure 9, the error (0.37 m) at the end of the field test run, at the Mojave Desert, is less than 1.5% of the distance traveled (29 m).

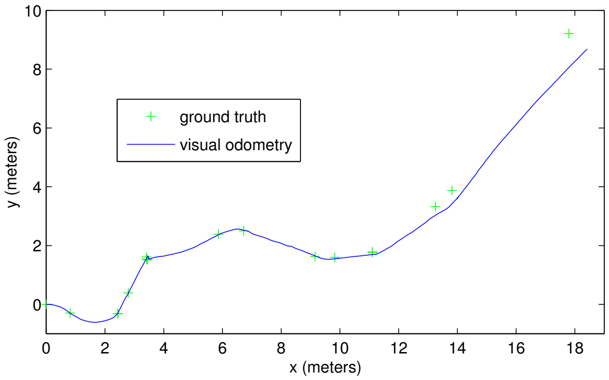

Figure 8: Mars Yard Run 2 Visual Odometry Results.

7.2 Slip-Compensated Path Follower Results

Results of the slip-compensated path following algorithm are shown in Figures 10 and 11. The entire section of the path shown in Figure 10 was on a slope of between 10° and 15°. Figure 10 is an expansion of the box shown in Figure 9 (note that the axes of Figure 10 have been rotated in order to minimize the vertical size of the plot). Figure 11 is an expansion of the box shown in Figure 10. These two figures show three important pieces of information that the slip-compensated path following algorithm uses to calculate the rover commands: visual odometry pose (as described in Section 2), kinematics pose (as described in Section 3), and the desired path. From this information the slip-compensated path follower calculates the rover commands as described in Section 6. Carrot heading, which is calculated in an intermediate step, is also shown. In Figure 10, the rover was able to accurately and efficiently follow the desired path, despite significant slippage. As can be seen in Figure 11, there is a noticeable bias between the visual odometry pose and the kinematics pose in the y direction. This is due to the downhill slippage of the rover; this bias is being compensated for by the slip compensation algorithm, as is evident from the fact that the rover accurately follows the desired path.

7.3 Continuous Motion Results

Results from the continuous motion slip compensation experiments are shown in Figure 12 (see Section 7.2 for a description of the elements of this plot). The top graph in the figure shows a 4.5 meter run using continuous slip compensation on a 10° slope (the slope is going down towards the positive y-axis). Again, notice the constant offset between the kinematic and visual odometry estimates, indicating slippage. The bottom graph shows an identical run without using slip compensation. As can be seen, the significant slippage shown in the bottom graph was compensated for in the run shown in the top graph. The mean square path offset error is 0.05 and 0.54 meters^2 for the runs with and without slip compensation, respectively. This is a significant improvement in performance.

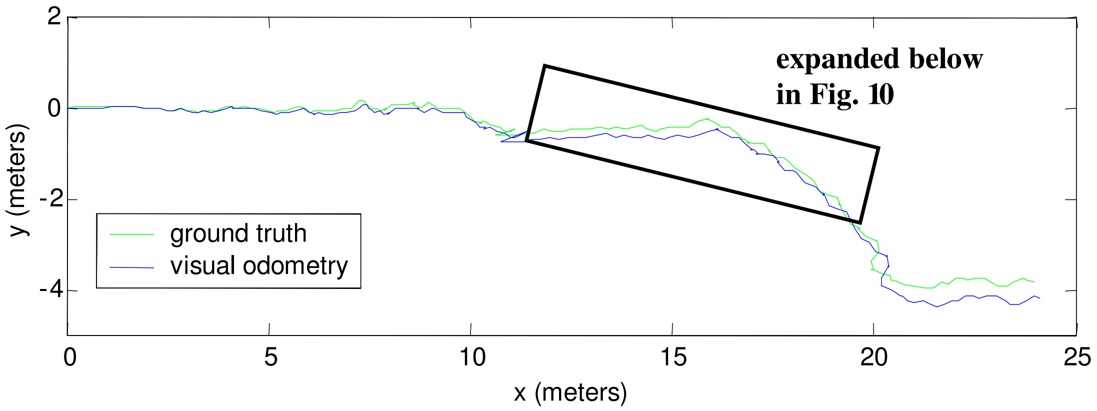

Figure 9: Field Test Visual Odometry Results.

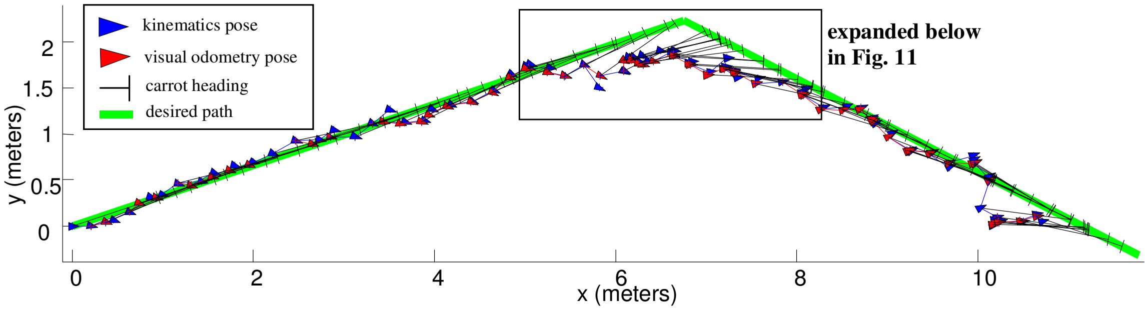

Figure 10: Field Test Slip Compensation Results.

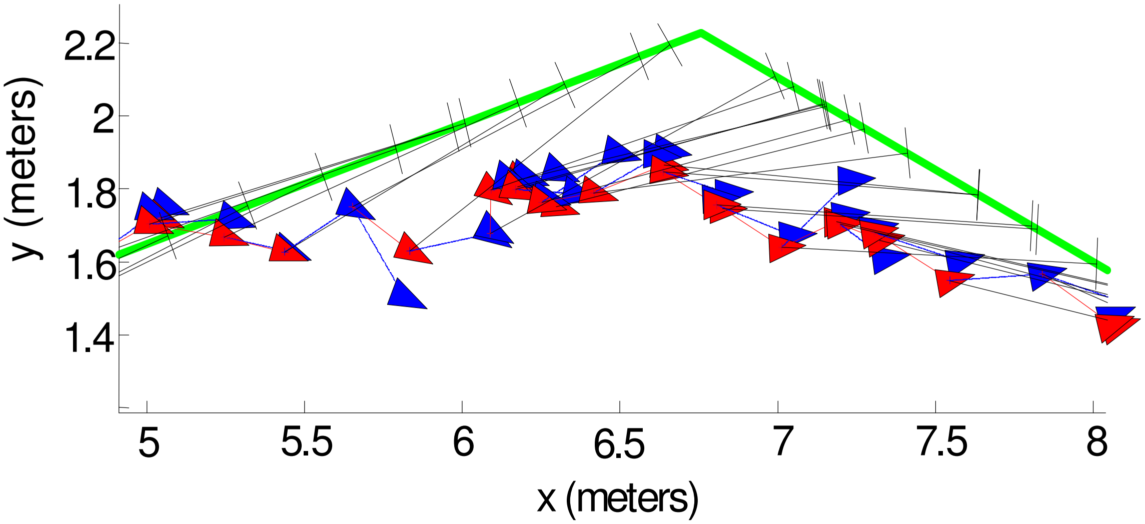

Figure 11: Expanded Slip Compensation Results.

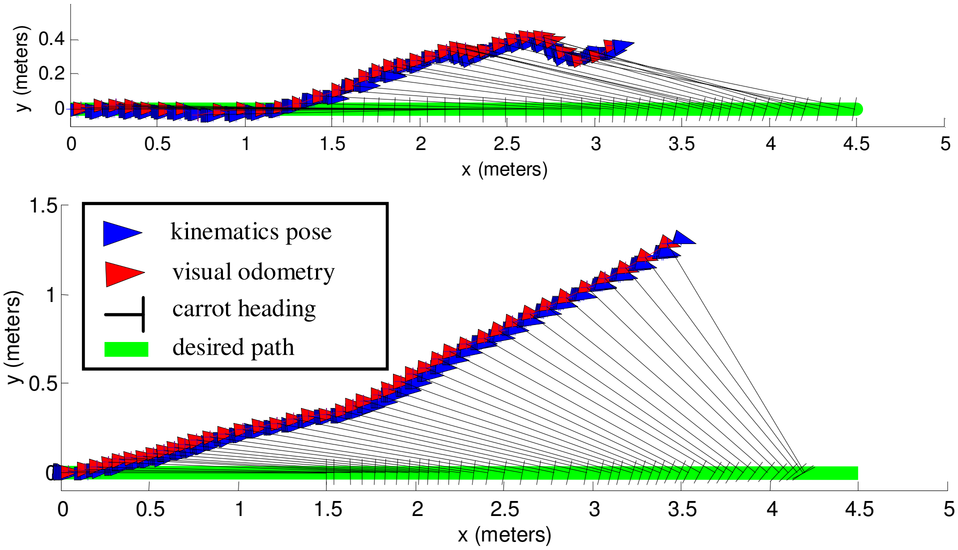

Figure 12: Tiltable Platform Slip Compensation Results (top: slip compensation enabled; bottom: slip compensation disabled).

8 CONCLUSIONS

In this paper, we have described the design, implementation, and testing of a system that enables a rover to accurately follow a designated path, compensate for slippage, and reach intended goals, independent of terrain geometry and soil characteristics along the path (within the mechanical constraints of the mobility system). Individual components have been simulated and tested; additionally, an integrated system has been tested onboard a rover in a desert field test. The results from the individual and integrated tests are encouraging. Visual odometry is able to consistently estimate rover motion to within 2.5% of distance traveled. Given this knowledge, the slip-compensated path following algorithm is able to accurately estimate and effectively compensate for slip and thus accurately follow a desired path and reach the intended goal while traversing through a high-slip environment. The algorithm has also been extended to allow for continuous motion slip compensation. Future work includes integration of this slip-compensated path following algorithm with a path planning/obstacle avoidance system such as Morphin or GESTALT [28, 29].

9 ACKNOWLEDGEMENTS

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract from the National Aeronautics and Space Administration (NASA), with funding from the Mars Technology Program, NASA Science Mission Directorate. The work of S.I. Roumeliotis was supported by the Jet Propulsion Laboratory (Grant No. 1248696), and the National Science Foundation (ITR-0324864, MRI-0420836).

REFERENCES

[1] D. Helmick, S.I. Roumeliotis, Y. Cheng, D. Clouse, M. Bajracharya, and L. Matthies, *Slip Compensation for a Mars Rover*, In Proc. 2005 IEEE International Conference on Intelligent Robots and Systems, Edmonton, Canada, Aug. 2-6, 2005, pp. 1419-1426.

[2] D. M. Helmick, Y. Chang, S. I. Roumeliotis, D. Clouse, and L. Matthies, *Path Following using Visual Odometry for a Mars Rover in High-Slip Environments*, In Proc. of the 2004 IEEE Aerospace Conference, Big Sky, MT, Mar. 6-13.

[3] Alonzo Kelly and Bryan Nagy, *Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control*, The International Journal of Robotics Research, Vol. 22, No. 7-8, 583-601 (2003).

[4] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, R. Petras, and S. Laubach, *The Rocky 7 Rover: A Mars Sciencecraft Prototype*, Proc. of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, April 20-25, 1997.

[5] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev, *Rocky 7: a next generation Mars rover prototype*, Advanced Robotics, Vol. 11, No. 4, pp. 341-358 (1997).

[6] P. Schenker, et. al., *Reconfigurable Robots for All Terrain Exploration*, Proceedings of SPIE, Vol. 4196, October, 2000.

[7] R. Hogg, A. Rankin, S. Roumeliotis, M. McHenry, D. Helmick, C. Bergh, and L. Matthies, *Algorithms and Sensors for Small Robot Path Following*, In Proc. 2002 IEEE International Conference on Robotics and Automation, Washington D.C., May 11-15.

[8] L. Matthies, *Dynamic Stereo Vision*, PhD thesis, Carnegie Mellon University, October 1989.

[9] C. F. Olson, L.H. Matthies, M. Shoppers, and M. Maimone, *Robust stereo ego-motion for long distance navigation*, Proc. of the IEEE Conference in Computer Vision and Pattern Recognition, Vol. 2. 2000.

[10] C. F. Olson, L.H. Matthies, M. Shoppers, and M. Maimone, *Stereo ego-motion Improvements for robust rover navigation*, Proc. of the 2001 IEEE International Conference on Robotics & Automation.

[11] Yang Cheng, Mark Maimone, Larry Matthies, *Visual Odometry on the Mars Exploration Rovers*, IEEE Conference on Systems, Man and Cybernetics, The Big Island, Hawaii, USA, October 2005.

[12] R. Deriche and G. Giraudon, *A computational approach for corner and vertex detection*, Int'l J. of Computer Vision Vol. 10, No 2. pp 101-124, 1993.

[13] P.H. Schonemann, *A generalized solution of the orthogonal Procrustes problem*, Psychometrika, 31:1-10, 1966.

[14] P. J. Rousseeuw, *Least median-of-squares regression*, Journal of the American Statistical Association, 79:871–880, 1984.

[15] M. Tarokh, G. McDermott, S. Hayati, and J. Hung, *Kinematic Modeling of a High Mobility Mars Rover*, Proc. of the IEEE International Conference on Robotics & Automation, May 1999.

[16] M. Tarokh, G. McDermott, and J. Hung, *Kinematics and Control of Rocky 7 Mars Rover*, Preliminary Report, Dept. of Math & Computer Sciences, San Diego State University, August 1998.

[17] P. F. Muir and C. P. Neumann, *Kinematic Modeling of Wheeled Mobile Robots*, Journal of Robotics Systems, Vol. 4, No. 2, pp. 281-340, 1987.

[18] D. Bickler, *A New Family of JPL Planetary Surface Vehicles*, In Missions, Technologies, and Design of Planetary Mobile Vehicles, pages 301-306, Toulouse, France, September 28-30, 1992.

[19] J. J. Craig, *Introduction to Robotics*, 2nd Ed., (Reading, Massachusetts: Addison Wesley), 1989.

[20] B. Friedland, *Analysis strapdown navigation using quaternions*, IEEE Transactions on Aerospace and Electronic Systems, AES-14(5): 764-768, Sep. 1978.

[21] E. J. Lefferts and F. L. Markley, *Dynamics modeling for attitude determination*, AIAA Paper 76-1910, Aug. 1976.

[22] S. I. Roumeliotis, *A Kalman filter for processing 3-D relative pose measurements*, Technical report, Robotics Laboratory, California Institute of Technology, Sep. 2001.
http://robotics.caltech.edu/~stergios/tech_reports/relative_3d_kf.pdf

[23] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, *Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization*, In Proc. of IEEE International Conference on Robotics and Automation, Detroit, MI, May 1999.

[24] S. I. Roumeliotis, *Robust Mobile Robot Localization: From single-robot uncertainties to multi-robot interdependencies*, PhD thesis, Electrical Engineering Department, University of Southern California, Los Angeles, CA, May 2000.

[25] S.I. Roumeliotis and J.W. Burdick, *Stochastic Cloning: A generalized framework for processing relative state measurements*, In Proc. 2002 IEEE International Conference on Robotics and Automation, Washington D.C., May 11-15, pp. 1788-95.

[26] A. Kelly, *A Feedforward Control Approach to the Local Navigation Problem for Autonomous Vehicles*, Robotics Institute Technical Report, CMU-RI-TR-94-17, Carnegie Mellon University, 1994.

[27] S. Singh et. al, *FastNav: A System for Fast Navigation*, Robotics Institute Technical Report CMU-RI-TR-91-20, Carnegie Mellon University, 1991.

[28] S. B. Goldberg, M. W. Maimone, and L. H. Matthies, *Stereo Vision and Rover Navigation Software for Planetary Exploration*, In 2002 IEEE Aerospace Conference Proc., Vol. 5, Big Sky, MT, March 2002.

[29] S. Singh, K. Schwehr, R. Simmons, T. Smith, A. Stentz, V. Verma, and A. Yahja, , *Recent progress in local and global traversability for planetary rovers*, In Proc. of International Conference on Robotics and Automation, 2000.