Slip Compensation for a Mars Rover

Daniel M. Helmick, Yang Cheng, Daniel S. Clouse,
Max Bajracharya, and Larry H. Matthies

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

firstname.lastname@jpl.nasa.gov

Stergios I. Roumeliotis

University of Minnesota
Dept. of Computer Science and Engineering4-192 EE/CS
200 Union Street SE, Minneapolis, MN 55455

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 compensation/path follower. Visual odometry tracks distinctive scene features in stereo imagery to estimate rover motion between successively acquired stereo image pairs. The vehicle kinematics for a rocker-bogie suspension system estimates motion by measuring wheel rates, and rocker, bogie, and steering angles. The Kalman filter merges data from an Inertial Measurement Unit (IMU) and visual odometry. This merged estimate is then compared to the kinematic estimate to determine how much slippage has occurred, taking into account estimate uncertainties. If slippage has occurred then a 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.**

Index Terms – rover navigation, visual odometry, slip compensation, kalman filter, kinematics.

I. Introduction

This paper describes the design, implementation, and experimental results of an integrated system that enables a Mars rover to navigate in a high slip environment. This system enables the rover to accurately follow a designated path, compensate for slippage, and reach intended goals independent of the terrain over which it is traversing (within the mechanical constraints of the mobility system). The system is comprised of several key components that were developed and refined 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 compensation/path following algorithm. Fig. 2 provides a high-level functional block diagram of the system. Visual odometry is an algorithm that uses stereo imagery to estimate rover motion independent of mechanical terrain properties and is described in *Section II*. The full vehicle kinematics, described in *Section III*, uses position sensor inputs from the joints and wheels of the rocker-bogie mobility system (see Fig. 1) to estimate rover motion. The Kalman filter merges estimates from visual odometry and the onboard IMU to estimate rover motion at high sample rates, and is detailed in *Section IV*. Because both the IMU estimate and the visual odometry estimate are independent of the vehicle's interaction with its

Figure 1: Rocky 8 on a Sandy Slope

environment, the motion estimate from the Kalman filter can be compared with the motion estimate from the vehicle kinematics, which is 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 estimate can contribute to the Kalman filter motion estimate. If, however, slippage has occurred, then the kinematic estimate and the Kalman filter estimate are differenced, resulting in a rover 'slip vector.' This slip vector is then used in combination with a path following algorithm to calculate rover velocity commands that follow a path while compensating for slip. This algorithm is described in greater detail in *Section V*.

The individual components of the system as well as a simplified integrated system has been tested onboard a rover. Several independent tests were performed using Rocky 8 (see Fig. 1), a Mars rover research platform. In the first test, visual odometry was tested onboard the rover in the JPL Mars Yard over two 25m traverses. Under normal conditions, wheel odometry accuracy is not better than 10% of distance traveled and, in higher slip environments, it can be significantly worse. Results from our first tests showed that we can achieve greater than 2.5% accuracy 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 done on a tiltable platform measuring 5x5 meters. These tests were done using continuous rover slip compensation algorithms (described in *Section V*), instead of stopping the rover to estimate slippage, as done in previous tests. Results from the three sets of experiments are provided in *Section VI*.

This paper extends the work described in [1]. Related work includes rover trajectory generation [2], rough terrain navigation [3], and path following [4].
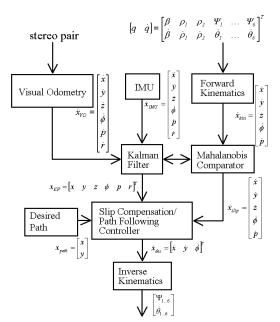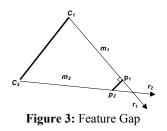
$$[q \quad \dot{q}] = \begin{bmatrix} \beta & \rho_1 & \rho_2 & \Psi_1 & \cdots & \Psi_6 \\ \dot{\beta} & \dot{\rho}_1 & \dot{\rho}_2 & \dot{\theta}_1 & \cdots & \dot{\theta}_6 \end{bmatrix}^T$$

Figure 2: Path Following/Slip Compensation System

II. VISUAL ODOMETRY ALGORITHMS

Mobile robot long distance navigation on a distant 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 [5]. Following this work, some minor variations and modifications were suggested for improving its robustness and accuracy [6,7]. The key idea of this method is to determine the change in position and attitude by solving for the transformation between a selection of 3D features extracted from consecutive stereo images by using maximum likelihood estimation. The basic steps of this method are described below. A more detailed description of this derivation can be found in [1].

A. *Feature Selection*

The first step in the visual odometry algorithm is to select features that can be easily matched between a stereo pair and tracked between image steps. To achieve this, the Forstner interest operator [8] is applied to the left image of the first stereo pair. The pixels with lower interest values are better features. In order to ensure that features are evenly distributed across the image scene, a minimum distance between any two features is enforced. In order to reduce the volume of data that needs to be sorted, the image scene is divided into grids, with a grid size significantly smaller than the minimum distance between features. Only the pixel with the lowest interest value in each grid is selected as a potential feature. Then, all potential features are sorted in descending order and the top N pixels meeting the minimum distance constraint are selected as features.

Figure 3: Feature Gap

B. *Feature Gap Analysis and Covariance Computation*

The 3D positions of the selected features are then 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 Fig. 3), r_1 and r_2 are two unit rays from the same feature in both images. Because of noise, r_1 and r_2 do not always intersect in space. The stereo point is taken 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

$$P_1 = C_1 + r_1 m_1$$
$$P_2 = C_2 + r_2 m_2 \tag{1}$$

where m_1 and m_2 are the length of $P_1 C_1$ and $P_2 C_2$. Therefore, we have

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

Then we have

$$P = (P_1 + P_2) / 2 \tag{3}$$

Taking the partial derivatives results in

$$P' = (r_1' m_1 + r_1 m_1' + r_2' m_2 + r_2 m_2') / 2.0 \tag{4}$$

The covariance of P is

$$\Sigma_p = P' \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} P'^t \tag{5}$$

where P' is the Jacobian matrix, or the matrix of first partial derivatives of P with respect to C_1 and C_2.

C. 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 location. Stereo matching is then performed on these tracked features on the second pair to determine their new 3D positions. If the initial motion guess is accurate, the difference between the two estimated 3D positions should be within the error ellipse of the previous covariance matrix.

D.   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 done in two steps. Coarse motion is first estimated with Schonemann motion estimation, and then a more accurate motion-estimate is determined by maximum likelihood motion estimation.

Schonemann motion estimation [9] uses singular value decomposition (SVD) with an orthogonal constraint to estimate a rotation matrix and a translation 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 method [10] is applied. 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 calculated. 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 estimation found using the above procedure is refined using maximum likelihood motion estimation. Maximum likelihood motion estimation takes account of the 3D position differences and associated error models in order to estimate motion. Let Q_{pj} and Q_{cj} be the observed feature positions before and after a robot motion. Then we have

$$Q_{cj} = RQ_{pj} + T + e_j \qquad (6)$$

where R and T are the rotation and translation of the robot and e_j is the combined errors in the observed positions of the jth features. In this estimation, the 3 axis rotations (Θ) and translation (T) are directly determined by minimizing the summation in the exponents $\sum e_j^T W_j e_j = min$, where $e_j = Q_{cj} - RQ_{pj} - T$ and W_j is the inverse covariance

matrix of e_j. The minimization of the nonlinear problem is done by linearization and iterations [5]. Two nice properties of maximum-likelihood estimation make the algorithm significantly more accurate than 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). Secondly, it incorporates error models in the estimation, which greatly improves the accuracy.

III. KINEMATIC ALGORITHMS

Full rover kinematic algorithms were developed to fill two roles in the system shown in Fig. 2. The first role is the forward kinematics of the vehicle, which estimates rover motion given the wheel rates and rocker, bogie, and steering angles. The second role is the inverse kinematics of the vehicle, which calculates 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 Fig. 1), but the techniques used to derive the algorithms could be used for any vehicle configuration (although there may be a fewer number of observable DOFs for different configurations). Additionally, these forward kinematic algorithms could be used directly for rovers with a subset of functionality (e.g. a rocker-bogie rover with only 4 steerable wheels, such as the Mars Exploration Rovers) simply by making the relevant parameters constant.

The motivation for developing the full kinematics of this class of vehicles (rather than making the more common planar assumption) is twofold. First, it allows for the observation of 5 DOFs, whereas the planar assumption limits this to 3 DOFs. Second, as terrain becomes rougher, the errors due to the planar assumption grow. These errors can grow to be significantly large (up to 30% 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 [11,12], with significant extensions being made for 6 wheel steering. Greater details of the kinematic derivations can be found in [1,11,12,13].

A.   Rocker-Bogie Configuration

The rocker-bogie configuration is a suspension system that is commonly used for planetary rovers and their prototypes. The configuration analyzed here consists of 15 DOFs: 6 steerable/drivable wheels (12 DOFs), a rocker, and two bogies. It is beyond the scope of this paper to discuss the benefits of such a mobility system [14]. What is relevant here is that with a few 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). However, these slip assumptions are only made for the kinematics algorithm (not for the slip compensation system as a whole); in fact, these assumptions, in part, are what enable the calculation of slip by the system.

B. D-H Table Formulation

Denavit-Hartenburg conventions were used to define the frames of each of the 15 DOFs. From the frame definitions, a unique set of D-H parameters can be derived that completely describes the kinematics of the rover. From these parameters, wheel Jacobians can be derived as described in [1].

C. Forward Kinematics

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

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

where $\dot{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{p} & \dot{r} \end{bmatrix}^T$ (the vector of rover velocities), $\dot{\eta}$ is a 6x1 vector of unobservable parameters, A is a 24x10 matrix of the unsensed elements [11,13] of the Jacobians, J_{comp} is a 24x16 block diagonal composite matrix of the wheel Jacobians, and $\dot{q}_{comp}$ is a 16x1 composite vector of measured kinematic rates.

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 it computationally much more efficient.

D. Inverse Kinematics

As can be seen in Fig. 2, inverse kinematics takes the commanded rover motion, and the current kinematic angles and angle rates as inputs, and produces six steering angles and six wheel rates. An interesting feature of the 6 steerable wheels is the fact that this creates a holonomic rover (with an assumption of instantaneous steering). Consequently all three controllable DOFs of the rover, $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}$, are independent, which allows for the isolation of several different control loops as will be seen in *Section V*.

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

$$x_O = \frac{\dot{y}_{cmd}}{\dot{\phi}_{cmd}} \qquad (8)$$

$$y_O = \frac{\dot{x}_{cmd}}{\dot{\phi}_{cmd}} \qquad (9)$$

This instantaneous center of rotation is then transformed from the rover frame (R) into each wheel frame (M_i) using

$$\begin{bmatrix} x_O \\ y_O \\ 0 \\ 1 \end{bmatrix}_{Mi} = T_R^{Mi} \cdot \begin{bmatrix} x_O \\ y_O \\ 0 \\ 1 \end{bmatrix}_R \quad \text{for } i = 1,\dots,6 \quad (10)$$

The steering angle is then calculated using

$$\Psi_i = a\,tan\left(\frac{x_{O,Mi}}{y_{O,Mi}}\right) \qquad (11)$$

Each wheel rate is then determined using

$$\dot{\theta}_i = \left[J_{ai}^T \Delta(J_{ui}) J_{ai} \right]^{-1} J_{ai}^T \Delta(J_{ui}) E_i \dot{v}_{cmd} \quad (12)$$

$$\Delta(J_{ui}) = J_{ui}(J_{ui}^T J_{ui})^{-1} J_{ui}^T - I \qquad (13)$$

where J_{ai} and J_{ui} are the actuated and un-actuated Jacobians defined in [13]. Eq. (12) is the actuated inverse solution from Muir and Neumann [13].

Again, the matrix inversions can be algebraically simplified so that each wheel rate calculation is relatively simple and computationally efficient.

IV. 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 [17, 19, 20] 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 [16, 18], while the visual odometry and vehicle kinematics (only when no slippage has occurred), are employed for updating the state estimate and providing periodic corrections. The equations of the EKF for a nonlinear system are in [18].

A. System Propagation Model

The state vector of interest in this estimation problem is:

$$x^T = [q^T \ \vec{b}_g^T \ \vec{u}^T \ \vec{b}_\alpha^T \ \vec{p}^T]$$

where q is the quaternion that represents the attitude of the vehicle, $\vec{u}^T$ and $\vec{p}$ are the linear velocity and position of the rover, and $\vec{b}_g$ and $\vec{b}_a$ are the biases in the gyroscope and accelerometer signals. The corresponding error state vector is:

$$\Delta x^T = [\Delta\vec{\theta}^T \ \Delta\vec{b}_g^T \ \Delta\vec{u}^T \ \Delta\vec{b}_\alpha^T \ \Delta\vec{p}^T]$$

where $\Delta\vec{o} = o - \hat{o}$ is the difference (error) between the real value of a state $\vec{o}$ and its estimate $\hat{\vec{o}}$, and $\delta\vec{\theta}$ is determined based on the small angle approximation:

$$\delta\vec{q}^T = \begin{bmatrix} \frac{1}{2}\delta\vec{\theta}^T & 1 \end{bmatrix}, q = \delta q \otimes \hat{q}.$$

The continuous time equation for the error-state propagation is

$$\Delta\dot{x}(t) = F_c\Delta x(t) + G_c w(t) \qquad (14)$$

By discretizing Eq. (14) we obtain:

$$\Delta x_{k+1} = F_k\Delta x_k + G_k w_k \qquad (15)$$

The interested reader is referred to [18] for the details of the derivation of Eqs. (14) and (15).

B.   Relative Pose Measurement Model

In what follows, we assume that at time t_k the vehicle is at position $^G p(t_k)$ with (quaternion) attitude $^1_G q(t_k) = q_1$ and after m steps it has moved to position $^G p(t_{k+m}) = {}^G p_2$ with attitude $^G_1 C(q)$. 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} correspondingly.

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

$$\Delta\tilde{z}_{k+m} = \begin{pmatrix} \Delta z_p \\ \Delta\tilde{z}_q \end{pmatrix} = X\begin{pmatrix} \Delta z_p \\ \Delta z_q \end{pmatrix} =$$
$$= \Gamma\begin{pmatrix} D_1 & D_2 \end{pmatrix}\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \mathcal{X} n_r = H\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \tilde{n}_r \qquad (16)$$

where Γ, D_1, D_2, and X are defined in [18].

Both noise n_r and $\tilde{n}_r$ are assumed to be a zero-mean white noise Gaussian processes with

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

As is evident from Eq. (16), the relative pose measurement error is expressed in terms of the current $\Delta x_2 = \Delta x(t_{k+m})$ and the previous $\Delta x_1 = \Delta x(t_k)$ (error) state of the system. The Kalman filter state vector must therefore be appropriately augmented to contain both of these state estimates. 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 $\begin{bmatrix} t_k & t_{k+m} \end{bmatrix}$.

C.   Augmented-state propagation

If $\Delta x_{k/k}$ 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:

$$\Delta\bar{x} = \begin{bmatrix} \Delta x_{k/k}^T & \Delta x_{k/k}^T \end{bmatrix}^T$$

Since initially, at time t_k, both version of the estimate of the error contain the same account of the information, the covariance matrix for the augmented system would be:

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

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}, the second copy of the state estimate is propagated (during this interval only) while the first remains stationary. The propagation equation for the augmented system is:

$$\begin{bmatrix} \Delta x_1 \\ \Delta x \end{bmatrix}_{k+1/k} = \begin{bmatrix} I & 0 \\ 0 & F_{k+1} \end{bmatrix}\begin{bmatrix} \Delta x_1 \\ \Delta x \end{bmatrix}_{k/k} + \begin{bmatrix} 0 \\ G_{k+1} \end{bmatrix}\bar{w}_k$$

or

$$\Delta\bar{x}_{k+1/k} = \bar{F}_{k+1}\Delta\bar{x}_{k/k} + \bar{G}_{k+1}\bar{w}_k$$

where Δx_1 is the non-moving copy of the error state of the vehicle. The covariance of the augmented system is propagated and after m steps is:

$$\bar{P}_{k+m/k} = \begin{pmatrix} P_{kk} & P_{kk}\mathcal{F}^T \\ \mathcal{F}P_{kk} & P_{k+m/k} \end{pmatrix} \qquad (17)$$

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}.

D.   Mahalanobis Comparator (Slippage Estimation)

In this section, we describe 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, these measurements need to be validated. If significant wheel slippage has occurred, the residual for the relative pose measurement 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 squared distance

$$d_m^2 = \tilde{r}_{k+m}^T \tilde{S}^{-1}\tilde{r}_{k+m} \qquad (18)$$

where $\tilde{r}_{k+m}$ is the measurement residual and $\tilde{S}$ is the corresponding residual covariance matrix, described in the

previous section. In the case of a vehicle odometry measurement, the Mahalanobis squared distance follows a Chi-square distribution with five degrees of freedom. A sufficient test for validating vehicle odometry measurements $\mathcal{Z}_{k+m}$ is to require that these match the expected (estimated by the EKF) measurements $\hat{\mathcal{Z}}_{k+m}$ 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 $d_m^2 \leq t$, with $t = 11.07$. If this inequality does not hold, these measurements are discarded and wheel slippage is detected. In this case, the residual $\tilde{r}_{k+m}$ is provided to the slip compensation algorithm for appropriately modifying the rover commands.

V. SLIP COMPENSATION/PATH FOLLOWING

At the center of Fig. 2 are the slip compensation/path following algorithms. These two algorithms 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 take a 3x1 slip vector $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}_{slip}$, a 3x1 rover pose vector $\begin{bmatrix} x & y & \phi \end{bmatrix}_{pose}$, and a 2x1 desired path vector $\begin{bmatrix} x & y \end{bmatrix}_{path}$ and then outputs a 3x1 commanded rover velocity vector $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}_{cmd}$.

A. Carrot Heading Algorithm

The carrot heading algorithm takes the desired path and the current rover pose and calculates a desired heading, ϕ_{carrot}, and therefore is able to calculate the heading error, ϕ_{err} of the rover. This algorithm was chosen for its robustness to path error [21,22]. The desired path consists of a set of linear segments between waypoints; however, the waypoints can be spaced 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 with the desired path and calculating the direction of that intersection. The intersection point that is furthest along the path is always selected. The heading error is then calculated using

$$\phi_{err} = \phi_{carrot} - \phi_{pose} \qquad (19)$$

A large radius will tend to filter out small features of a path, but results in a smooth motion of the rover. A small radius results in large heading changes of the rover for small path errors (which is extremely inefficient), but results in an overall smaller path following error. 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.

B. Slip Compensation/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 has not occurred 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

$$\dot{\phi}_{cmd} = (K_1 \cdot \phi_{err} + K_2 \dot{\phi}_{slip}) / T_S \qquad (20)$$

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, $\dot{\phi}_{slip}$. It attempts to achieve the heading deemed by the carrot algorithm to be optimal, even when slipping in the yaw direction. The second loop is described by the equation

$$\dot{y}_{cmd} = K_3 \cdot \dot{y}_{slip} / T_S \qquad (21)$$

This loop calculates the rate of the rover in the y direction based entirely on the slip in the y direction during the previous sample period. A $\dot{y}$ command results in a crabbing maneuver, where all six wheels have a steering angle offset in the same direction. $\dot{x}_{cmd}$ 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 three rover commands, $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}_{cmd}$, are then passed to the inverse kinematics.

C. Continuous Motion

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

There are many benefits of continuous motion: more efficient motion (both in energy and time), less slippage due to smaller changes in momentum (this becomes more significant as the rover increases in mass and the terrain gets steeper), and it allows for a higher rate of slip compensation, thus enabling 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 allows more complex spline paths to be the basis of path planning. Another advantage is that it allows more optimal spacing between navigation stops (allowing for other requirements, such as IMU bias zeroing, to drive the spacing between stops).

VI. Results

Three sets of experiments have been performed using Rocky 8 (see Fig. 1), 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 rover. The body mounted hazard cameras on Rocky 8 have a resolution of 640x480 with a field of view of 79.5°x64.0°, a baseline of 8.4 cm, and are angled down at 45°. The first set of experiments was performed in the JPL Marsyard, a 20x20 meter space designed as an analog (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 consisted of slopes up to 25° consisting of loose granular sand (see Fig. 1). This set of experiments was a test of a simplified integrated slip compensation/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 whenever 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 5x5 meter 'sandbox' that could be tilted in discrete increments from 0° to 30°. 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 V* above.

In the first two sets of experiments, ground truth data was 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 to the rover (see Fig. 1) whenever 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 the runs.

A. Visual Odometry Results

Visual odometry results are shown from both the Marsyard and the Johnson Valley experiments. The results from both Marsyard runs are in Figs. 4 and 5. The errors at the end of the runs are both less than 2.5% of the distance traveled. As can be seen in Fig. 6, the error at the end of the field test run (0.37 m) is less than 1.5% of distance traveled (29 m).

B. Slip Compensation/Path Following Results

Results of the slip compensation/path following algorithm are shown in Figs. 7 and 8. The entire section of the path shown in Fig. 7 was on a slope of between 10° and 15°. Fig. 7 is an expansion of the box shown in Fig. 6. Fig.
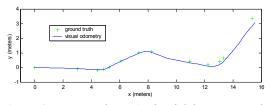

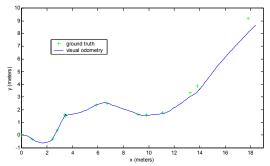**Figure 4:** Mars Yard Run 1 Visual Odometry Results

Figure 5: Mars Yard Run 2 Visual Odometry Results

8 is an expansion of the box shown in Fig. 7. These two figures show three important pieces of information that the slip compensation/path following algorithm uses to calculate the rover commands: visual odometry pose, kinematics pose, and the desired path. Carrot heading, which is calculated in an intermediate step, is also shown. In Fig. 7, the rover was able to accurately and efficiently follow the desired path, despite significant slippage. As can be seen in Fig. 8, there is a noticeable, consistent 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 in the slip compensation algorithm, as is evident because the rover accurately follows the desired path.

C. Continuous Motion

Results from the continuous motion slip compensation experiments are shown in Fig. 9. The top graph in the figure shows a 4.5 meter run using continuous slip compensation on a 10° slope. 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.

VII. 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 (minus the Kalman filter) 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
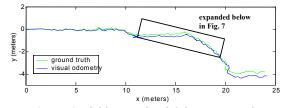
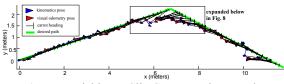**Figure 6:** Field Test Visual Odometry Results

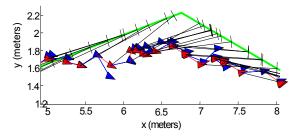**Figure 7:** Field Test Slip Compensation Results

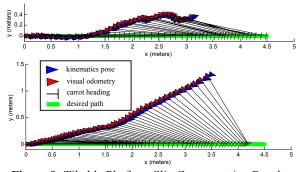**Figure 8:** Expanded Slip Compensation Results

Figure 9: Tiltable Platform Slip Compensation Results
(**top:** slip comp. enabled; **bottom:** slip comp. disabled)

within 2.5% of distance traveled. Given this knowledge, the slip compensation/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 path following/slip compensation algorithm with a path planning/obstacle avoidance system such as MORPHIN or GESTALT [23, 24].

References

[1] 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.

[2] A. Kelly and B. Nagy, "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control," submitted to the International Journal of Robotics Research, Summer 2002.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

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

[18] S. I. Roumeliotis, "A Kalman filter for processing 3-D relative pose measurements," Technical report, Robotics Laboratory, California Institute of Technology, Sep. 2001.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.