Path Following using Visual Odometry for a Mars Rover in High-Slip Environments[1]

Daniel M. Helmick, Yang Cheng,
Daniel S. Clouse, and Larry H. Matthies
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
818-354-3226
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
612-626-7507
stergios@cs.umn.edu

Abstract— An architecture for autonomous operation of Mars rovers in high slip environments has been designed, implemented, and tested. This architecture is composed of several key technologies that enable 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). These technologies include: visual odometry, full vehicle kinematics, a Kalman filter, 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 using a maximum likelihood motion estimation algorithm. The full vehicle kinematics for a rocker-bogie suspension system estimates motion, with a no-slip assumption, 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 (taking into account estimate uncertainties) if and how much slippage has occurred. If no statistically significant slippage has occurred then the kinematic estimate is used to complement the Kalman filter estimate. 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, in conjunction with the inverse kinematics, to determine the necessary wheel velocities and steering angles to compensate for slip and follow the desired path.

TABLE OF CONTENTS

1. INTRODUCTION

There is a very strong scientific rationale for Mars rover exploration on slopes, to access channels, layered terrain, and putative shorelines and fluid seeps, in pursuit of evidence for fossil or extant life and in order to understand the geologic and climatic history of the planet. Precision landing capabilities anticipated for 2009 and beyond will bring such terrain within practical reach of rover missions

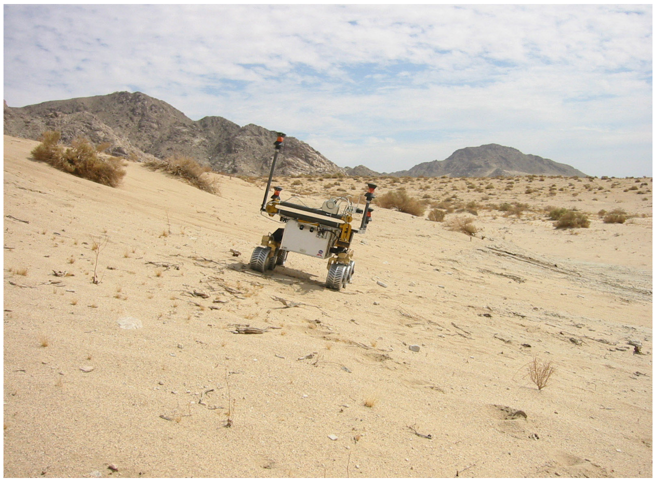

Figure 1: Rocky 8 on a Sandy Slope

[1] 0-7803-8155-6/04/$17.00© 2004 IEEE

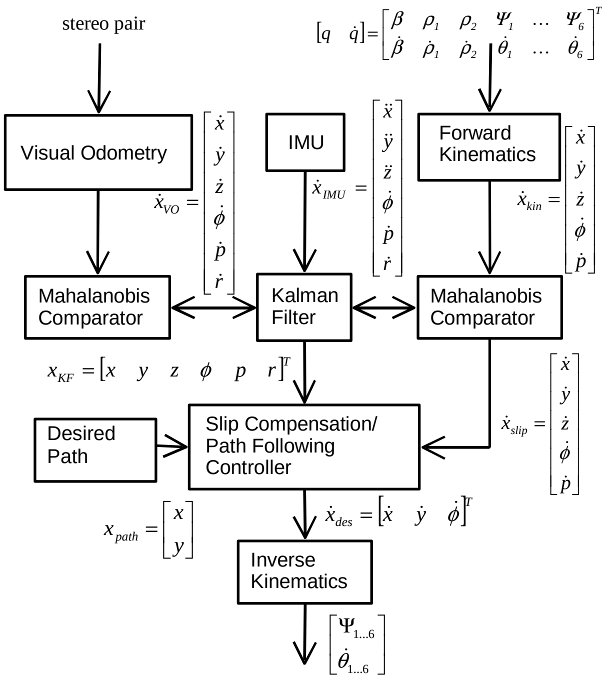
stereo pair
$[q \quad \dot{q}] = \begin{bmatrix} \beta & \rho_1 & \rho_2 & \Psi_1 & \dots & \Psi_6 \\ \dot{\beta} & \dot{\rho}_1 & \dot{\rho}_2 & \dot{\theta}_1 & \dots & \dot{\theta}_6 \end{bmatrix}^T$
Visual Odometry
IMU
Forward Kinematics
$\dot{x}_{VO} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix}$
$\dot{x}_{IMU} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix}$
$\dot{x}_{kin} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \end{bmatrix}$
Mahalanobis Comparator
Kalman Filter
Mahalanobis Comparator
$x_{KF} = [x \quad y \quad z \quad \phi \quad p \quad r]^T$
$\dot{x}_{slip} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{p} \end{bmatrix}$
Desired Path
Slip Compensation/ Path Following Controller
$x_{path} = \begin{bmatrix} x \\ y \end{bmatrix}$
$\dot{x}_{des} = [\dot{x} \quad \dot{y} \quad \dot{\phi}]^T$
Inverse Kinematics
$\begin{bmatrix} \Psi_{1\dots6} \\ \dot{\theta}_{1\dots6} \end{bmatrix}$

Figure 2: Path Following/Slip Compensation Architecture

for the first time. MOLA results [1] show Valles Marineris slopes over 600 meter baselines of typically less than 5 degrees on lower canyon walls to 28 degrees on upper walls, while slopes elsewhere on Mars are within this upper bound. MOLA measurements of inner slopes of craters with seeps is typically from 15 to over 21 degrees for baselines of 330 meters, though typical fresh-looking Mars craters have slopes 2 to 2.5 times lower [2]. In principle, such slopes are within the mobility envelope of rovers. However, such slopes are likely to have abundant loose material, which could cause significant wheel slippage and sinkage in addition to the usual obstacles posed by rocks and tip-over hazards. Relatively little research has been done on rover navigation on slopes or in the presence of significant wheel slippage or sinkage, particularly for transverse slippage when traveling across a slope. Therefore, there is little rover navigation experience or autonomous navigation capability for what is likely to be important terrain for future rover missions.

This paper describes the design, implementation, and experimental results of an integrated system to enable navigation of Mars rovers in high slip environments. 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 architecture 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, and a slip compensation/path following algorithm. A high-level functional block diagram of the system can be seen in Figure 2. Visual odometry is an algorithm that uses stereo imagery to estimate rover motion independent of terrain properties. The full vehicle kinematics 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 merges estimates from visual odometry and the onboard IMU to estimate rover motion at very high rates. Both the IMU estimate and the visual odometry estimate are independent of the vehicle's interaction with its environment. The motion estimate from the Kalman filter is then compared with the motion estimate from the vehicle kinematics, which is highly dependent upon the vehicle's interaction with its environment. This comparison allows for the statistical analysis of the difference between the two estimates. Accounting for estimate uncertainties, if the vehicle kinematic motion estimate can contribute to the Kalman filter motion estimate, then no statistically significant slippage has occurred. 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.

The individual components of the system as well as that of a simplified integrated system has been tested onboard a rover. Two independent tests were performed using Rocky 8 (see Figure 4), 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. Results from this visual odometry test are very encouraging. 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 tests show that we can achieve greater than 2.5% accuracy, regardless of the mechanical soil characteristics. The second test was a field test that used the slip compensation architecture described above, minus the Kalman filter. This test involved traverses of over 50 meter on sandy slopes.

Results from the Mars Yard test and the field test are presented in Section 6. Testing of the fully integrated system, including the Kalman filter and the continuous slip compensation, is planned for the near future.

2. 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. Techniques for position estimation by stereo sequences have been shown to be a very reliable and accurate method. Visual odometry, or ego-motion estimation, was originally developed by Matthies [3]. Following this work, some minor variations and modifications were suggested for improving its robustness and accuracy [4][5]. However, the key idea of this method remains the same: to determine the change in

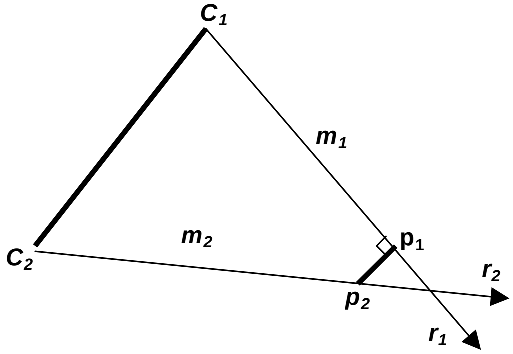

Figure 3: Feature Gap

position and attitude for two or more pairs of stereo images by using maximum likelihood estimation. The basic steps of this method are described below.

Feature Selection

Features that can be easily matched between a stereo pair and tracked between image steps are selected. The Forstner interest operator [6] 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. The top N pixels meeting the minimum distance constraint are selected as features.

Feature Gap Analysis and Covariance Computation

The 3D positions of the selected features are determined by stereo matching. 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.

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), $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are two unit rays from the same feature in both images. Because of noise, $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ 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 $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$, thus, we have

$$\begin{aligned} P_1 &= C_1 + r_1 m_1 \\ P_2 &= C_2 + r_2 m_2 \end{aligned} \qquad (1)$$

where m_1 and m_2 are the length of P_1C_1 and P_2C_2.

Therefore, we have

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

Then we have

$$m_1 = \frac{Br_1 - (B\cdot r_2)(r_1\cdot r_2)}{1-(r_1\cdot r_2)^2} \quad m_2 = (r_1\cdot r_2)m_1 - Br_2 \quad (3)$$

$$P = (P_1 + P_2)/2.0 \qquad (4)$$

where $B = C_2 - C_1$ and m_1 and m_2 are functions of feature locations on both images. Taking the partial derivatives results in

$$\begin{aligned} m_1' = {} & \frac{[B\cdot r_1' - (B\cdot r_2')\cdot G - (B\cdot r_2)\cdot F][1-G^2]}{[1-G^2]^2} + \\ & \frac{2[B\cdot r_1 - (B\cdot r_2)\cdot G][(G\cdot F)]}{[1-G^2]^2} \end{aligned} \qquad (5)$$

$$m_2' = G\cdot m_1' + F\cdot m_1 - B\cdot r_2' \qquad (6)$$

$$P' = (r_1' m_1 + r_1 m_1' + r_2' m_2 + r_2 m_2')/2.0 \qquad (7)$$

where $F = (r_1'\cdot r_2 + r_1\cdot r_2')$ and $G = (r_1\cdot r_2)$.

The covariance of P is

$$\Sigma_p = P' \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} P'^t \qquad (8)$$

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

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. Then a correlation-based search and tracking based on an affine template precisely determine these features' 2D positions in the second image pair. The affine template tracking aims to remove the tracking error caused by large roll and scale change between images. In this case, the relationship between two images within the template is expressed as an affine transform

$$x_2 = ax_1 + by_1 + c \qquad y_2 = dx_1 + ey_1 + f \tag{9}$$

Where [a, b, c, d, e, f] are the unknown coefficients of the affine transform that can be determined by an iterative method by minimizing a merit function [7]

$$M = \sum [I_1(x_1, y_1) - I_2(x_2, y_2)]^2 = min \tag{10}$$

where $I_j(x, y)$ specifies the pixel value at position (x, y) in image j. 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. However, when the initial motion guess is off, the difference between the two estimated positions reflects the error of the initial motion guess and can be used to determine the change of rover position.

Motion Estimation

Motion estimation 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 [8] 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, the least-median-of-squares method [9] 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

$$Q_{cj} = RQ_{pj} + T + e_j \tag{11}$$

where R and T are the rotation and translation of the robot and e_i 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 $\sum r_j^T W_j r_j = \min$, where $r_j = Q_{cj} - RQ_{pj} - T$ and W_j is the inverse covariance matrix of e_j. The minimization of the nonlinear problem is done by linearization and iterations [3]. Two nice properties of maximum-likelihood estimation make the algorithm powerful. 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.

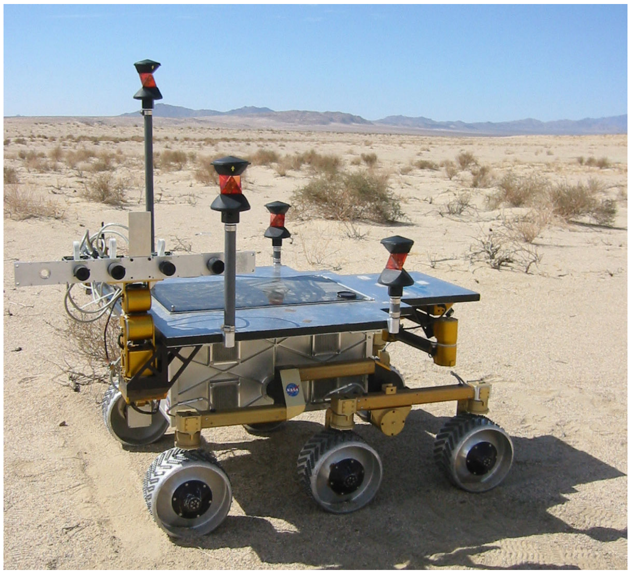

Figure 4: Rocky 8

3. Kinematic Algorithms

Full rover kinematic algorithms were developed to fill two roles in the architecture shown in Figure 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 to create the desired rover motion.

These algorithms are specific to the rocker-bogie configuration with six steerable wheels (see Figure 4), 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, the 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 MER) simply by making the relevant parameters constant.

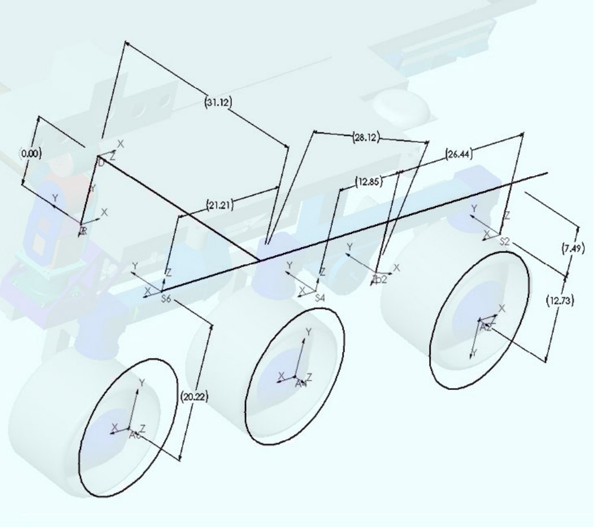

Figure 6: Coordinate Frame Definition For Right Side of Rover (all dimensions in cm)

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. As shown in Section 6, these errors can grow to be significantly large and affect the slip calculations and, consequently, the slip compensation controller. Secondarily, no formulation of the inverse kinematics previously existed that would take advantage of the holonomic nature of the rover (see *Inverse Kinematics* below).

The formulation of the forward and inverse kinematics closely follows that of [10,11], with significant extensions being made for 6 wheel steering. Greater details of the kinematic derivations can be found here [10,11,12].

Also note that the implementation of these algorithms has been parameterized so that changes in the kinematics of the rover or application of this algorithm to a different rocker-bogie rover does not require a re-derivation of the algorithms and can be made with only parameter changes in code.

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. The interested reader is referred to [13] for more details. 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).

The first assumption increases the modeling error in rough terrain; however, this error still remains smaller than the error created by the planar assumption. The second assumption is necessary for a steerable wheel to exist.

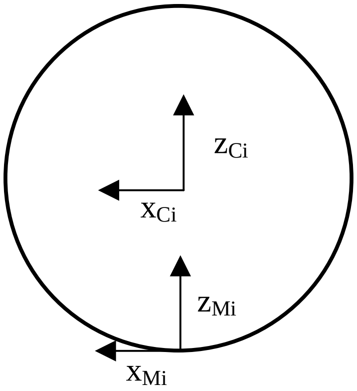

Figure 5: Contact Frame and Motion Frame Definitions

Frame Definitions

Denavit-Hartenburg conventions were used to define the frames of each of the 15 DOFs (see Figure 6) [14]. Two additional coordinate frames were added for each wheel: the contact frame, C_i and the motion frame, M_i (for i=1,2,…,6). C_i defines the wheel/terrain contact point, and M_i defines the steering slip and the wheel roll (see Figure 5). Table 1 describes each frame.

Table 1: Frame Descriptions

Frame Identification	Frame Description
R	rover frame
D	rocker (differential) frame
ρ_1	right bogie frame
ρ_2	left bogie frame
S_1,…,S_6	steering frames for each wheel
A_1,…,A_6	axel frames for each wheel
C_1,…,C_6	contact frames for each wheel
M_1,…,M_6	motion frames for each wheel

D-H Table Formulation

From the frame definitions a unique set of D-H parameters can be derived that completely describes the kinematics of the rover (see Table 2 in Appendix A). From these parameters, wheel Jacobians can be derived, as described in the following section.

Wheel Jacobian Generation

Homogeneous transforms can be derived for each wheel using

$$T^{R}_{Mi} = T^{R}_{D} \cdot T^{D}_{\rho 1,2} \cdot T^{\rho 1,2}_{Si} \cdot T^{Si}_{Ai} \cdot T^{Ai}_{Ci} \cdot T^{Ci}_{Mi} \qquad (12)$$

(for the two rear wheels not connected to the bogies replace the $T^{D}_{\rho 1,2} \cdot T^{\rho 1,2}_{Si}$ with T^{D}_{Si}), where each of these transforms are functions of the current kinematic angle measurements. From these transforms and the equality

$$\dot{T}^{R}_{R} = T^{R}_{Mi} \cdot \dot{T}^{Mi}_{R} = \begin{bmatrix} 0 & -\dot{\phi} & \dot{p} & \dot{x} \\ \dot{\phi} & 0 & -\dot{r} & \dot{y} \\ -\dot{p} & \dot{r} & 0 & \dot{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

the wheel Jacobians can be calculated such that

$$\dot{v} = J_i \dot{q}_i \qquad i = 1,2,...,6 \qquad (14)$$

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, and $\dot{q}_i = \begin{bmatrix} \dot{\beta} & \dot{\rho}_{1,2} & \dot{\theta}_i \end{bmatrix}$, the vector of time derivative joint angles for each wheel.

Forward Kinematics (Least-Squares Motion Estimation)

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

$$\begin{bmatrix} \dot{v} \\ \dot{\eta} \end{bmatrix} = (A^T A)^{-1} A^T \cdot J_{comp} \dot{q}_{comp} \qquad (15)$$

where A is a 24x10 matrix, $\dot{\eta}$ is a 6x1 vector of unobservable parameters, 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 this inversion of A^TA onboard the rover. The matrix equations can be greatly simplified algebraically to make it computationally much more efficient. We were able to reduce total computation time of the forward kinematics algorithm down to 1.0 msec on an UltraSparc/300MHz.

Inverse Kinematics

As can be seen in Figure 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. Thus 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 5.

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

$$x_O = \frac{\dot{y}_{cmd}}{\dot{\phi}_{cmd}} \qquad (16)$$

$$y_O = \frac{\dot{x}_{cmd}}{\dot{\phi}_{cmd}} \qquad (17)$$

For $\dot{\phi}_{des} = 0$ (any type of pure crabbing maneuver, including straight line driving) x_O and y_O are infinite and not very useful, so the instantaneous center of rotation is defined in polar coordinates using

$$\theta_O = -a\tan\left(\frac{\dot{y}_{cmd}}{\dot{x}_{cmd}}\right) \qquad (18)$$

$$r_O = \infty \qquad (19)$$

This instantaneous center of rotation is then transformed from the rover frame into each motion frame using

$$\begin{bmatrix} x_O \\ y_O \\ 0 \\ 1 \end{bmatrix}_{Mi} = T^{Mi}_{R} \cdot \begin{bmatrix} x_O \\ y_O \\ 0 \\ 1 \end{bmatrix}_{R} \quad \text{for i = 1,…,6} \qquad (20)$$

The steering angle is then calculated using

$$\Psi_i = a\tan\left(\frac{x_{O,Mi}}{y_{O,Mi}}\right) \qquad (21)$$

or

$$\Psi_i = -\theta_O \tag{22}$$

The wheel rate calculation is started by splitting up equation 14 into actuated and un-actuated components

$$E_i \dot{v}_{cmd} = J_{ai}\dot{\theta}_i + J_{ui}\dot{q}_{ui} \quad \text{for i = 1,…,6} \tag{23}$$

Each wheel rate is then determined using

$$\dot{\theta}_i = \left[J_{ai}^T \Delta(J_{ui}) J_{ai}\right]^{-1} J_{ai}^T \Delta(J_{ui}) E_i \dot{v}_{cmd} \tag{24}$$

$$\Delta(J_{ui}) = J_{ui} (J_{ui}^T J_{ui})^{-1} J_{ui}^T - I \tag{25}$$

Equation 24 is the actuated inverse solution from Muir and Neumann [12].

Again, the 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 (kinematic algorithms). 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 [16, 18, 19] 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 [15, 17], while the odometry, visual and vehicle, are employed for updating the state estimate and providing periodic corrections. The equations of the EKF for a nonlinear system are listed in Appendix B.

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:

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

The continuous time equation for the error-state propagation is

$$\Delta\dot{x}(t) = F_c \Delta x(t) + G_c w(t) \tag{26}$$

with

$$F_c = \begin{pmatrix} -\lfloor \hat{\vec{\omega}}_m \rfloor & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -{}^G_1C^T(q)\lfloor \hat{\vec{\alpha}}_m \rfloor & 0 & 0 & -{}^G_1C^T(q) & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{pmatrix},$$

$$G_c = \begin{pmatrix} -I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -{}^G_1C^T(q) & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$w^T = \left[\vec{n}_r^T \ \vec{n}_w^T \ \vec{n}_g^T \ \vec{n}_u^T\right]$$

where ${}^G_1C^T(q)$ is the transpose of the rotational matrix from the current attitude {1} to the global frame of the reference {G}, $\vec{n}_r^T, \vec{n}_w^T, (\vec{n}_g^T \ \vec{n}_u^T)$ are the noise vectors due to the white-noise and bias components of the gyroscopes (accels), $\lfloor \hat{\vec{\omega}}_m \rfloor = \lfloor \vec{\omega}_m \rfloor - \lfloor \hat{\vec{b}}_g \rfloor$, $\lfloor \hat{\vec{\alpha}}_m \rfloor = \lfloor \vec{\alpha}_m \rfloor - \lfloor \hat{\vec{b}}_\alpha \rfloor - \lfloor {}^G_1C^T(q)\vec{g} \rfloor$, $\vec{\omega}_m, \vec{\alpha}_m$ are the rotational velocity and linear acceleration as measured by the gyroscopes and the accelerometers, ${}^G\vec{g}$ is the gravitational acceleration and

$$\lfloor \vec{o} \rfloor = \begin{pmatrix} 0 & -o_3 & o_2 \\ o_3 & 0 & -o_1 \\ -o_2 & o_1 & 0 \end{pmatrix}.$$

By discretizing Eq. (26) we obtain:

$$\Delta x_{k+1} = F_k \Delta x_k + G_k w_k \tag{27}$$

The interested reader is referred to [17] for the details of the derivation of Eqs. (26), (27) and the specific form of the matrices F_k and G_k used in Eq. (51) for propagating the covariance of the state estimate.

Measurement Update Model

In the following three sections we derive the equations of the EKF that processes relative pose measurements from odometry (visual and/or vehicle) in order to update the estimate for the state of the rover. 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.

Relative Position Measurement Error

The relative position measurement z_p between the two locations {1}, and {2} can be written as:

$$z_p = {}^1p_2 + n_p = {}^G_1C^T(q)\left({}^G p_2 - {}^G p_1\right) + n_p \qquad (28)$$

where n_p is the noise associated with this measurement assumed to be a zero-mean white Gaussian process with covariance $R_p = E\left[n_p n_p^T\right]$. ${}^G_1C(q)$ is the rotational matrix that expresses the orientation transformation between frames {G} and {1}.

If Δp_i is the error in the estimate of the position p_i and the δq is the error in the estimate of the attitude q then:[2]

$$p_i = \Delta p_i + \hat{p}_i,\, i = 1,2,\ldots\; q = \delta q \otimes \hat{q}$$

Equation (28) can now be written as:

$$z_p = {}^G_1C^T(\delta q \otimes \hat{q})\left({}^G\hat{p}_2 + \Delta p_2 - {}^G p_1 - \Delta p_1\right) + n_p \qquad (29)$$

The estimated relative position measurement is:

$$\hat{z}_p = {}^G_1C^T(\hat{q})\left({}^G\hat{p}_2 - {}^G p_1\right) = {}^G_1C^T(\hat{q})\,{}^G\hat{p}_{1,2} \qquad (30)$$

The error in the relative position measurement is:

$$\Delta z_p = z_p - \hat{z}_p$$

By substituting from Eqs. (29), (30) and employing the small error angle approximation

$$\delta q \simeq \left[\delta\vec{q}\;\; 1\right]^T \simeq \left[\frac{1}{2}\vec{\theta}_1\;\; 1\right]^{\mathrm{T}},$$

it can be shown [17] that:

$$\begin{aligned}\Delta z_p \simeq {}&{}^G_1C^T(\hat{q}_1)\left\lfloor {}^G\hat{p}_{1,2}\right\rfloor \delta\vec{\theta}_1 + \\ &+ {}^G_1C^T(\hat{q}_1)\Delta p_2 - {}^G_1C^T(\hat{q}_1)\Delta p_1 + n_p\end{aligned}. \qquad (31)$$

In Eq. (31) the first term expresses the effect of the orientation uncertainty at the time t_k on the quality of the estimated measurement. Note that if at time t_k there was no uncertainty about the orientation of the vehicle that would mean $\delta\vec{\theta}_1 = 0$ and thus the error in the relative position measurement would depend only on the errors in the estimates of the previous and the current position of the vehicle.

Relative Attitude Measurement Error

The relative attitude measurement error between the two locations {1}, and {2} is:

$$\Delta z_q = z_q - \frac{1}{2}\hat{q} = \frac{1}{2}q + n_q - \frac{1}{2}\hat{q} \qquad (32)$$

Where n_q is the relative attitude measurement noise. We assume that n_q is a zero – mean white Gaussian process with covariance $R_q = E\left[n_q {n_q}^T\right]$. Since

$$\frac{1}{2}q = {}^1_0q \otimes {}^2_0q^{-1} = q(t_k) \otimes q^{-1}(t_{k+m}) = q_1 \otimes q_2^{-1}$$

and

$$q_i = \delta q_i \otimes \hat{q}_i,\, i = 1,2$$

${}^1_2q$ can be written as:

$${}^1_2q = \delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1} \qquad (33)$$

By substituting Eq. (33) in Eq. (32) we have:

$$\Delta z_q = \delta q_1 \otimes {}^1_2\hat{q} \otimes \delta q_2^{-1} - \frac{1}{2}\hat{q} + n_q \qquad (34)$$

[2] Note that from here on q refers to q_1 and δq refers to δq_1. We have also dropped the vector symbol from the real, measured, estimated, and error position to simplify notation.

With

$$\delta q_1 = \begin{bmatrix} \delta\vec{q}_1 \\ \delta q_{1_4} \end{bmatrix}, \delta q_2 = \begin{bmatrix} \delta\vec{q}_2 \\ \delta q_{2_4} \end{bmatrix}, \delta q_2{}^{-1} = \begin{bmatrix} -\delta\vec{q}_2 \\ \delta q_{2_4} \end{bmatrix}$$

In order to simplify the notation we set:

$$\frac{1}{2}\hat{q} = q = \begin{bmatrix} \vec{q} \\ q_4 \end{bmatrix}$$

For small attitude estimation errors δq_1 and δq_2 we make the following approximations: $\delta q_{2_4} \simeq 1$, $\delta q_{2_4} \simeq 1$, $\delta\vec{q}_1 \ll 1_{3\times 1}$, $\delta\vec{q}_2 \ll 1_{3\times 1}$. The first term in Eq. (34) can be written as:

$$\delta q_1 \otimes {}_2^1\hat{q} \otimes \delta q_2^{-1} \simeq \begin{pmatrix} \vec{q} + q_4\left(\delta\vec{q}_1 - \delta\vec{q}_2\right) + \lfloor \vec{q} \rfloor \left(\delta\vec{q}_1 + \delta\vec{q}_2\right) \\ q_4 - \vec{q}^T\left(\delta\vec{q}_1 - \delta\vec{q}_2\right) \end{pmatrix} \quad (35)$$

By multiplying both sides of Eq. (32) with the matrix

$$\Xi^{\mathrm{T}}\left({}_2^1\hat{q}\right) = \Xi^{\mathrm{T}}(q) = \left[\left(q_4 I - \lfloor \vec{q} \rfloor - \vec{q}\right)\right] \quad (36)$$

we define the vector attitude measurement error as:

$$\begin{aligned} \Delta\tilde{z}_q &= \Xi^T\left({}_2^1\hat{q}\right)\Delta z_q = \Xi^T\left({}_2^1\hat{q}\right)\left({}_2^1 q + n_q\right) - 0 \\ &= \Xi^T\left({}_2^1\hat{q}\right)\left(\delta q_1 \otimes {}_2^1\hat{q} \otimes \delta q_2^{-1}\right) + \Xi^T\left({}_2^1\hat{q}\right) n_q \end{aligned} \quad (37)$$

By substituting from Eqs. (35), (36) the first term in the previous equation can be written as:

$$\Xi^T\left({}_2^1\hat{q}\right)\left(\delta q_1 \otimes {}_2^1\hat{q} \otimes \delta q_2^{-1}\right) \simeq \delta\vec{q}_1 - {}_2^1C\left({}_2^1\hat{q}\right)\delta\vec{q}_2 \quad (38)$$

Eq. (37) is now expressed as:

$$\Delta\tilde{z}_q \simeq \frac{1}{2}\left(\delta\vec{\theta}_1 - {}_2^1C\left({}_2^1\hat{q}\right)\delta\vec{\theta}_2\right) + \tilde{n}_q \quad (39)$$

where we have used the small angle approximation $\delta\vec{q}_i = \frac{1}{2}\vec{\theta}_i, i = 1,2$ and $\tilde{n}_q = \Xi^T\left({}_2^1\hat{q}\right) n_q$ with

$$\tilde{R}_q = E\left[\tilde{n}_q\tilde{n}_q^T\right] = \Xi^T\left({}_2^1\hat{q}\right) R_q \Xi\left({}_2^1\hat{q}\right).$$

Relative Pose Measurement Error

The Indirect Kalman filter estimates the error in: (i) attitude $\delta\vec{\theta}$, (ii) gyroscopes biases $\Delta\vec{b}_\alpha$, (iii) velocity $\Delta\vec{u}$, (iv) accelerometers biases $\Delta\vec{b}_a$, and (v) position $\Delta\vec{p}$. The error state vectors estimated by the filter at times t_k and t_{k+m} for i=1, 2 are:

$$\Delta x_i = \left[\delta\vec{\theta}_i^T \;\; \Delta\vec{b}_{gi}^T \;\; \Delta\vec{u}_i^T \;\; \Delta\vec{b}_{\alpha i}^T \;\; \Delta\vec{p}_i^T\right]^T$$

The errors in the relative position and attitude (pose) measurements calculated in Eqs. (31) and (39) are:

$$\begin{aligned} \Delta\tilde{z}_{k+m} &= \begin{pmatrix} \Delta z_p \\ \Delta\tilde{z}_q \end{pmatrix} = \mathrm{X}\begin{pmatrix} \Delta z_p \\ \Delta z_q \end{pmatrix} = \\ &= \Gamma\left(D_1 \quad D_2\right)\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \mathcal{X} n_r = H\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \tilde{n}_r \end{aligned} \quad (40)$$

with

$$\Gamma = \begin{pmatrix} {}_1^GC^T\left(\hat{q}_1\right) & 0 \\ 0 & {}_1^GC^T\left(\hat{q}_1\right) \end{pmatrix}$$

$$D_1 = \begin{pmatrix} \lfloor {}^G\hat{p}_{1,2} \rfloor & 0 & 0 & 0 & -I \\ \left(\frac{1}{2}\right){}_1^GC\left(\hat{q}_1{}^{-1}\right) & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & I \\ \left(-\frac{1}{2}\right){}_2^GC\left(\hat{q}_2{}^{-1}\right) & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{X} = \begin{pmatrix} I_{3x3} & 0_{3x4} \\ 0_{3x3} & \Xi^T\left({}_2^1\hat{q}\right) \end{pmatrix}$$

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

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

As is evident from Eq. (41), 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 $\left[t_k \quad t_{k+m}\right]$.

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:

$$\Delta\breve{x} = \left[\Delta x_{k/k}^T \; \Delta x_{k/k}^T\right]^T$$

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:

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

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}, the second copy of the state estimate is propagated (during this interval only) while the first remains stationary[3]. The propagation equation for the augmented system based on Eq. (46) is:

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

or

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

where Δx_1 is the non-moving copy of the error state of the vehicle. The covariance of the augmented system is propagated according to Eq. (51) and after m steps is:

$$\breve{P}_{k+m/k} = \begin{pmatrix} P_{kk} & P_{kk}F^{\mathrm{T}} \\ FP_{kk} & P_{k+m/k} \end{pmatrix} \tag{41}$$

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}.

State and Covariance Update Equations

When the relative pose measurement is received the covariance matrix for the residual is given by Eq. (52):

$$S = H\breve{P}_{k+m/k}H^T + \tilde{R}_r \tag{42}$$

where $\tilde{R}_r = XR_rX^T$ is the adjusted covariance for the relative pose measurement and R_r is the initial covariance of this noise as calculated by the odometry algorithm. We define the pseudo-residual covariance matrix as $\tilde{S} = \Gamma^{-1}S\Gamma$ and by substituting from Eqs. (41), (42):

$$\begin{aligned} \tilde{S} = {} & D_1P_{kk}D_1{}^T + D_2\mathcal{F}P_{kk}D_1{}^T + D_1P_{kk}\mathcal{F}^TD_2{}^T + \\ & + D_2P_{k+m/k}D_2{}^T + \mathcal{R}_r \end{aligned}$$

where $\mathcal{R}_r = \Gamma^{-1}\tilde{R}_r\Gamma$. The update covariance matrix is calculated from the Eq. (54) as:

$$\begin{aligned} \breve{P}_{k+m/k+m} &= \breve{P}_{k+m/k} - \breve{P}_{k+m/k}H^TS^{-1}H\breve{P}_{k+m/k} \\ &= \breve{P}_{k+m/k} - \begin{bmatrix} P_{kk}D_1^T + P_{kk}\mathcal{F}^{\mathcal{T}}D_2^T \\ \mathcal{F}P_{kk}D_1^T + P_{k+m/k}D_2^T \end{bmatrix}\tilde{S}^{-1}\times \\ & \begin{bmatrix} D_1P_{kk} + D_2\mathcal{F}P_{\mathrm{kk}} + D_1P_{kk}\mathcal{F}^{\mathrm{T}} + D_2P_{k+m/k} \end{bmatrix} \end{aligned} \tag{43}$$

The update covariance matrix for the new state of the vehicle will be (lower-right diagonal submatrix):

$$\begin{aligned} \breve{P}_{k+m/k+m} &= \breve{P}_{k+m/k} - \\ &= (\mathcal{F}P_{\mathrm{kk}}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}(D_1P_{kk}\mathcal{F}^T + D_2P_{k+m/k}) \end{aligned}$$

The Kalman gain is calculated by applying Eq. (54):

$$K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}\breve{P}_{k+m/k}H^TS^{-1} \tag{44}$$

with

$$K_2 = (\mathcal{F}P_{\mathrm{kk}}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}\Gamma^T \tag{45}$$

The residual is calculated as in Eq. (56):

$$\tilde{r}_{k+m} = \Delta\tilde{z}_{k+m} = \mathcal{X}\Delta z_{k+m} = \begin{bmatrix} z_p - \hat{z}_p \\ \Xi^T({}^{1}_{2}\hat{q})(z_q - \hat{z}_q) \end{bmatrix}$$

where z_p, z_q are the relative position and orientation measurements provided by the odometry,

$$\hat{z}_p = {}^{1}\hat{p}_2 = {}^{G}_{1}C^T(\hat{q}_1)({}^{G}\hat{p}_2 - {}^{G}\hat{p}_1), \hat{z}_q = {}^{1}_{2}\hat{q}$$

and

$$\Xi^T({}^{1}_{2}\hat{q})\hat{z}_q = \Xi^T({}^{1}_{2}\hat{q})_2{}^{1}\hat{q} = 0_{3\times 1}$$

Thus

$$\tilde{r}_{k+m} = \begin{bmatrix} z_p - {}^{G}_{1}C^T(\hat{q}_1){}^{G}\hat{p}_{1,2} \\ \Xi^T({}^{1}_{2}\hat{q})\hat{z}_q \end{bmatrix}$$

Finally, the updated augmented state is given by Eq. (57):

$$\breve{x}_{k+m/k+m} = \breve{x}_{k+m/k} + K\tilde{r}_{k+m}$$

From Eq. (45) the (evolving) state will be updated as:

[3] In the derivation of the equations of the Kalman filter that processes relative pose measurements, we duplicated the state estimate and its corresponding covariance at time t_k and allowed each of them to evolve separately. We have coined the term *stochastic cloning* for this new technique.

$$x_{k+m/k+m} = x_{k+m/k}(\mathcal{F}P_{\mathrm{kk}}D_1^T + P_{k+m/k}D_2^T)\tilde{S}^{-1}\times (\mathcal{Z} + \hat{\mathcal{Z}}_{k+m/k})$$

where

$$\mathcal{Z}_{k+m/k} = \begin{bmatrix} {}_1^GC^T(\hat{q}_1^{-1})z_p \\ {}_1^GC^T(\hat{q}_1^{-1})z_p\Xi^T({}_2^1\hat{q})z_q \end{bmatrix}, \hat{\mathcal{Z}}_{k+m} = \begin{bmatrix} {}^G\hat{p}_{1,2} \\ 0 \end{bmatrix}$$

is the pseudo-measurement of the relative displacement (pose) expressed in global coordinates. The quantities $\hat{q}_1^{-1}, {}_2^1\hat{q} = {}_G^1\hat{q} \otimes {}_G^2\hat{q}^{-1} = \hat{q}_1 \otimes \hat{q}_2^{-1}$ and $\hat{p}_{1,2} = {}^G\hat{p}_2 - {}^G\hat{p}_1$ are computed using the previous and current state estimates from the filter. Note that the current state estimates at time t_{k+m} are calculated by propagating the previous state estimates at time t_k using the rotational velocity and linear acceleration measurements from the IMU.

The same process is repeated every time a new set of relative pose measurements $z(t_{k+\lambda m}) = \begin{bmatrix} z_p^T(t_{k+\lambda m}) & z_q^T(t_{k+\lambda m}) \end{bmatrix}^T$, $\lambda = 1, 2, \ldots$ becomes available.

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

$$d_m^2 = \tilde{r}_{k+m}^T\tilde{S}^{-1}\tilde{r}_{k+m} \tag{46}$$

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 $d_m^2 \le t$, 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.

5. SLIP COMPENSATION/PATH FOLLOWING

At the center of Figure 2 are the slip compensation/path following algorithms. These two algorithms are used in close conjunction to achieve this architecture's end goal of enabling the traversal of a desired path through high slip environments, such as sandy slopes. 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 $\begin{bmatrix}x & y\end{bmatrix}_{path}$. It then outputs a 3x1 commanded rover velocity vector $\begin{bmatrix}\dot{x} & \dot{y} & \dot{\phi}\end{bmatrix}_{cmd}$.

Carrot Heading Algorithm

The carrot heading algorithm takes the desired path and the current rover pose and calculates a desired heading, ϕ_{carrot}, and thus is then able to calculate the heading error, ϕ_{err} of the rover (see Figure 7). This algorithm was chosen for its robustness to path error [20,21]. 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

$$\phi_{err} = \phi_{carrot} - \phi_{pose} \tag{47}$$

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.

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.

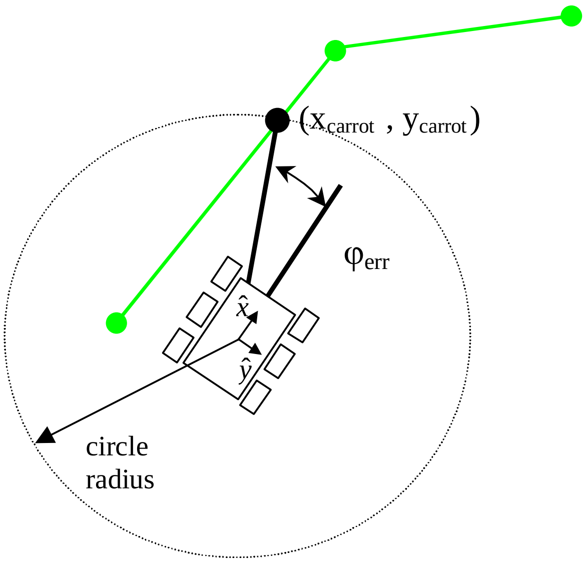

Figure 7: Carrot Heading Calculation

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

$$\dot{\phi}_{cmd} = (K_1 \cdot \phi_{err} + K_2 \dot{\phi}_{slip}) / T_S \qquad (48)$$

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

$$\dot{y}_{cmd} = K_3 \cdot \dot{y}_{slip} / T_S \qquad (49)$$

This loop calculates the rate of the rover in the y direction (perpendicular to normal straight line motion of the rover) 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).

Both of these loops make the assumption that the slip from the last sample period has some correlation to the slip in the current sample period. This assumption becomes more valid as the sample period decreases.

These three rover commands, $\begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}_{cmd}$, are then passed to the inverse kinematics algorithm.

6. Results

Two experiments have been performed using Rocky 8 (see 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 rover. The body mounted hazard cameras on Rocky 8 have a resolution of 640x480 with a field of view of 79.5x64.0 (horizontal x vertical) and a baseline of 8.4 cm and are angled down at 45˚. The first experiment was performed in the JPL's Marsyard, a 20mx20m space designed as an analog (in rock size/distribution and soil characteristics) to the Viking Lander sites. It consisted of two consecutive 25-meter runs with visual odometry running onboard. The second experiment was performed in Johnson Valley, California. The terrain of this area consisted of slopes of loose granular sand up to 25˚ (see Figure 1). This experiment 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, and a slip estimate was calculated and compensated for only when 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 effects of non-zero rocker and bogie angles are shown in simulation results.

In both experiments, ground truth data was collected with a Leica Total Station, which is a laser based position measurement system. The Total Station was used to measure the absolute position of four prisms mounted to the rover (see Figure 4). 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 Total Station and a prism. The prism would be placed somewhere within the range of the Total Station and surveyed. This location would then be transformed into the initial rover frame where it could be used directly as a waypoint.

Visual Odometry Results

Visual odometry results are shown from both the Marsyard and the Johnson Valley experiments. The results from both Marsyard runs are shown in Figures 8 thru 13. In Figures 8 and 9 the errors at the end of the runs are both less than 2.5% of the distance traveled. Errors in attitude (shown in Figures 10-13) remain below ~5˚ throughout both runs.

As can be seen in Figure 14, the error at the end of the field test run (0.37 m) is less than 1.5% of distance traveled (29 m).

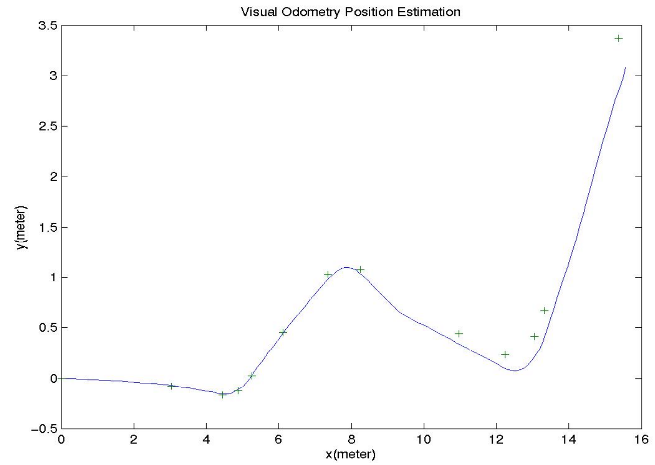

Figure 8: X and Y from Run 1 (ground truth -- +)

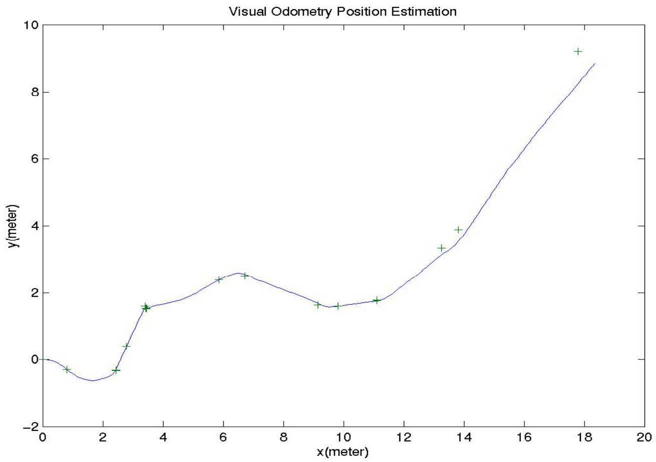

Figure 9: X and Y from Run 2 (ground truth -- +)

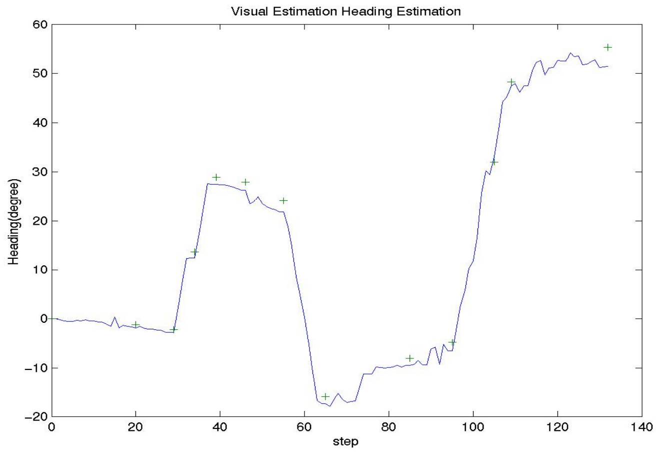

Figure 10: Heading from Run 1 (ground truth -- +)

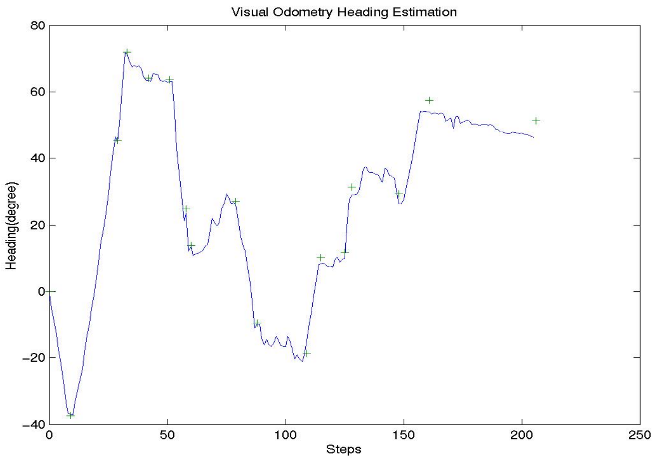

Figure 11: Heading from Run 2 (ground truth -- +)

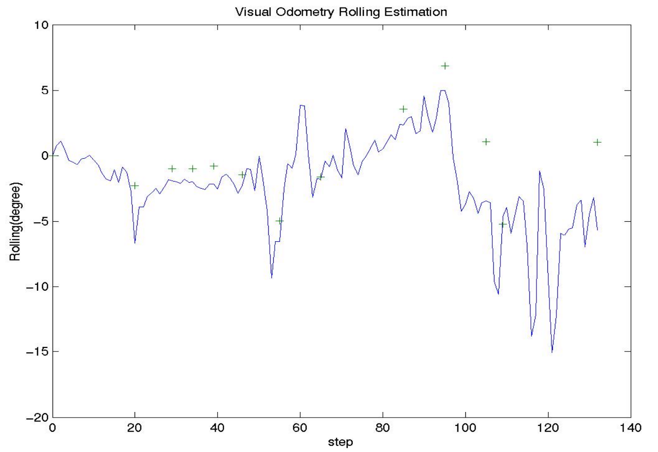

Figure 12: Roll from Run 1 (ground truth -- +)

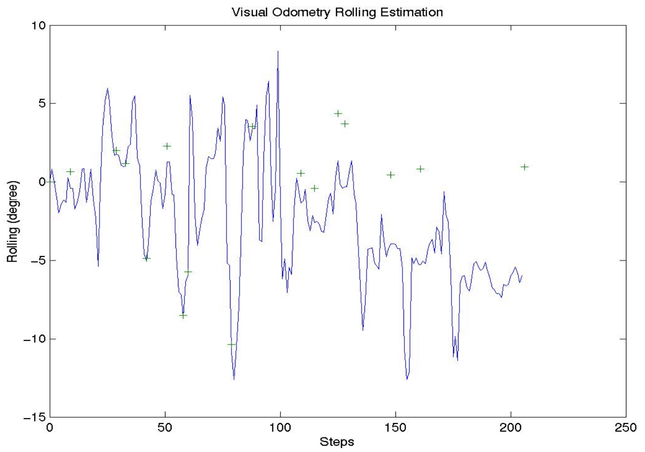

Figure 13: Roll from Run 2 (ground truth -- +)

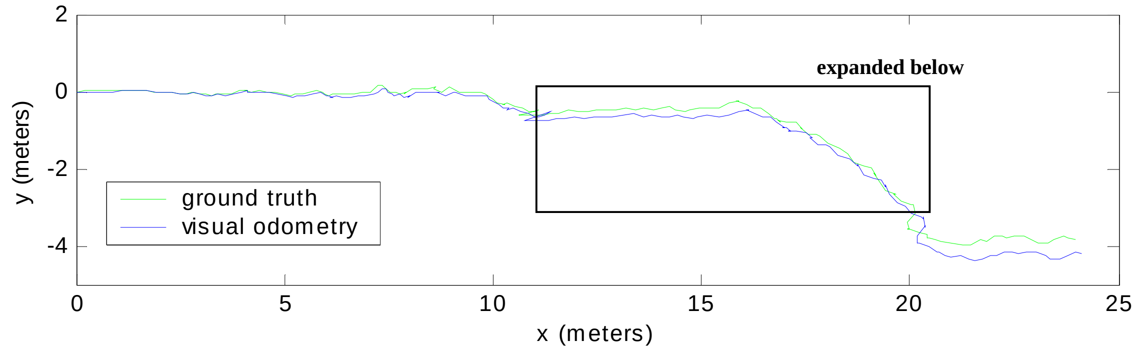

Figure 14: Field Test Visual Odometry Results

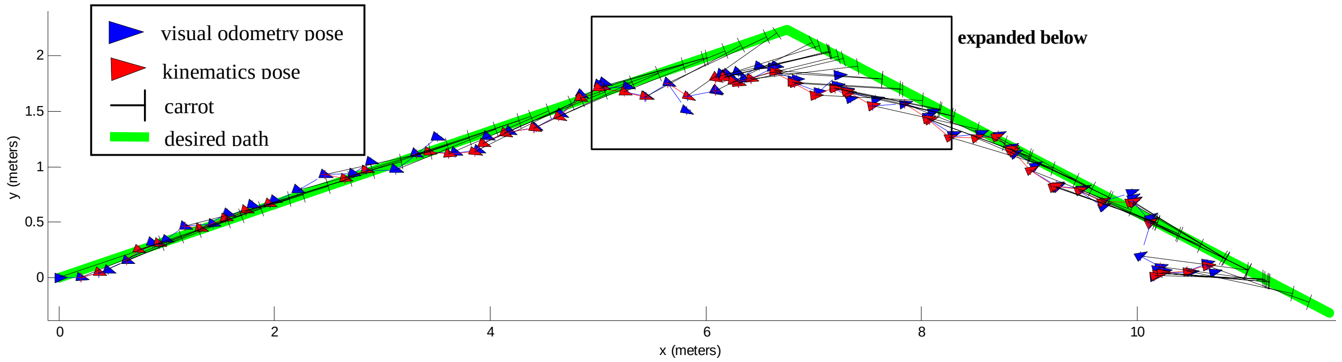

Figure 15: Field Test Slip Compensation/Path Following Results

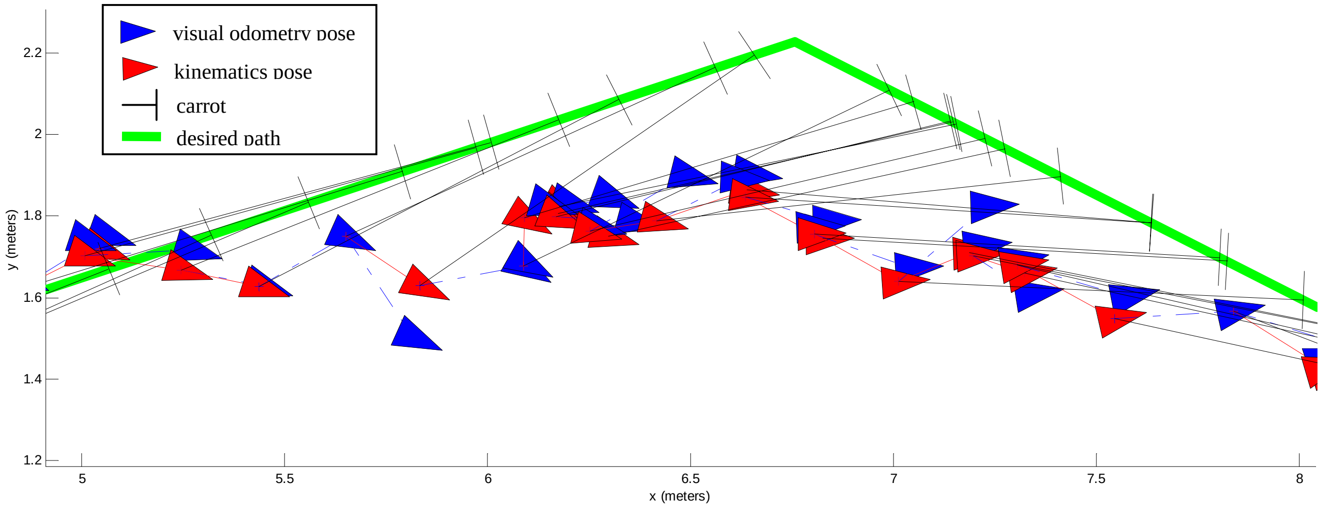

Figure 16: Expanded Slip Compensation/Path Following Results

Kinematic Simulation Results

The goal of this kinematic simulation was to compare kinematics that make the planar assumption (2D kinematics) with the full kinematics described in this paper. The simulation was of extremely rough terrain that exercised the rocker and the two bogies of the suspension system to their full extent. As can be seen in Figure 17, under such extreme conditions, very large errors can quickly accumulate. Over the short distance of the simulation (~ 0.15cm of travel in the x direction) the 2D kinematics had accumulated an error of greater than 30% distance traveled.

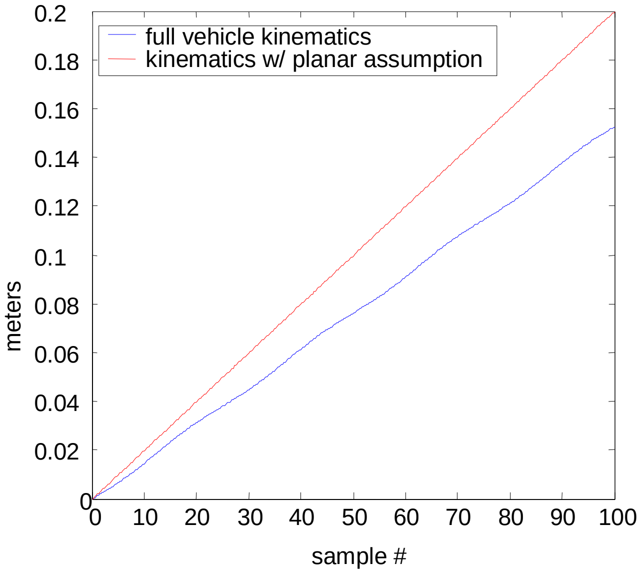

Figure 17: X Distance of Rover in World Frame

Slip Compensation/Path Following Results

Results of the slip compensation/path following algorithm are shown in Figures 15 and 16. The entire section of the path shown in Figure 15 was on a slope of between 10˚ and 15˚. Figure 15 is an expansion (and a rotation) of the box shown in Figure 14. Figure 16 is an expansion of the box shown in Figure 15. 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 Figure 15, the rover was able to accurately and efficiently follow the desired path, despite significant slippage. As can be seen in Figure 16, 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.

7. 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 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.

ACKNOWLEDGEMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors would like to thank two people who were instrumental in the achievement of this work. Max Bajracharya was critical to the success of the Johnson Valley field test. Siqi Chen spent a productive summer at JPL developing the automated ground truth data collection system that saved us both time and energy in the field.

APPENDIX A

Table 2: D-H Parameters for Rocky 8

Frame	γ (rad)	d (m)	a (m)	α (rad)
D	0	-0.2022	-0.1420	-π/2
ρ_1	β-0.2698	0.3111	0.2812	0
ρ_2	-β-0.2698	-0.3111	0.2812	0
S_1	ρ_1+0.2698	0	0.2644	-π/2
S_2	ρ_2+0.2698	0	0.2644	-π/2
S_3	ρ_1-2.872	0	0.1285	π/2
S_4	ρ_2-2.872	0	0.1285	π/2
S_5	β+π	0.3111	0.2121	π/2
S_6	-β+π	-0.3111	0.2121	π/2
A_1	Ψ_1	-0.1273	0	-π/2
A_2	Ψ_2	-0.1273	0	-π/2
A_3	Ψ_3	-0.1273	0	π/2
A_4	Ψ_4	-0.1273	0	π/2
A_5	Ψ_5	-0.2022	0	π/2
A_6	Ψ_6	-0.2022	0	π/2
C_1	0	0	0	π/2
C_2	0	0	0	π/2
C_3	π	0	0	π/2
C_4	π	0	0	π/2
C_5	π	0	0	π/2
C_6	π	0	0	π/2
M_1	ξ_1	-.1000	-.1000*θ_1	0
M_2	ξ_2	-.1000	-.1000*θ_2	0
M_3	ξ_3	-.1000	-.1000*θ_3	0
M_4	ξ_4	-.1000	-.1000*θ_4	0
M_5	ξ_5	-.1000	-.1000*θ_5	0
M_6	ξ_6	-.1000	-.1000*θ_6	0

APPENDIX B

Propagation

$$\Delta x_{k+1/k} = F_{k+1}\Delta x_{k/k} + G_{k+1}w_k \qquad (50)$$

$$P_{k+1/k} = F_{k+1}P_{k/k}F_{k+1}^T + G_{k+1}Q_kG_{k+1}^T \qquad (51)$$

Update

$$S = HP_{k+1/k}H^T + R \qquad (52)$$

$$K = P_{k+1/k}H^TS^{-1} \qquad (53)$$

$$P_{k+1/k+1} = P_{k+1/k} - P_{k+1/k}H^TS^{-1}HP_{k+1/k} \qquad (54)$$

$$r_{k+1} = z_{k+1} - \hat{z}_{k+1} = \Delta z_{k+1} \qquad (55)$$

$$x_{k+1/k+1} = x_{k+1/k} + Kr_{k+1} \qquad (56)$$

REFERENCES

[1] O. Aharonson, M. T. Zuber, G. A. Neumann, and J. W. Head, “Mars: Northern hemisphere slopes and slope distributions,” *Geophysical Research Letters*, 25(24), pp. 4413-4416, December 15, 1998.

[2] J. Garvin, Lead Scientist, NASA Mars Exploration Program, personal communication.

[3] L. Matthies. “Dynamic Stereo Vision,” PhD thesis, Carnegie Mellon University, October 1989.

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

[5] C. F. Olson, L.H. Matthies, M. Shoppers, and M. Maimone, “Stereo ego-motion Improvements for robust rover navigation,” Proceedings of the IEEE International Conference on Robotics & Automation, pages 1099-1104, 2001.

[6] 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.

[7] Richard Szeliski, “Video Mosaics for Virtual Environments,” IEEE Computer Graphics and Applications, March 1996.

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

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

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

[11] 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.

[12] 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.

[13] 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.

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

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

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

[17] 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

[18] 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 Proceedings of IEEE International Conference on Robotics and Automation, volume 2, pages 1656–1663, Detroit, MI, May 10-15 1999.

[19] Stergios 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.

[20] 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.

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

BIOGRAPHY

Daniel Helmick *received his B.S degree in Mechanical Engineering from Virginia Polytechnic Institute and State University and his M.S. in Mechanical Engineering with a specialization in controls from Georgia Institute of Technology in 1996 and 1999 respectively. Since June 1999 he has been working at the Jet Propulsion Laboratory in the Telerobotics Research and Applications Group on robotics research projects involving vision/sensor based control of robots, state estimation, and navigation and mobility algorithms. He has worked on robotic vehicles covering a wide range of functionality, including: Mars research rovers for rough terrain mobility; small, tracked robots for urban mobility; a cryobot for ice penetration; and reconfigurable wheeled robots for Mars exploration. His research interests include: sensor-based control of robots, sensor fusion and state estimation, and rover navigation and mobility.*

Yang Cheng *is a senior research staff member in the Machine Vision group of Jet Propulsion Laboratory. He eared his Ph.D. in remote sensing from the Geography Department of the University of South Carolina and was then a staff member at Oak Ridge National Laboratory. Since starting at JPL in 1999, he has worked on several space robotic research projects including Landmark Based Small Body Navigation System, Vision and Navigation Subsystem of the FIDO rover, and Passive Imaging Based Spacecraft Safe landing. His research interests include robotic navigational autonomy, computer vision, cartography, map projection, geographic information system (GIS), parallel computing, and more.*

Stergios Roumeliotis *received his diploma in electrical engineering from the National Technical University of Athens (NTUA), Greece in 1995, and his M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Southern California (USC), Los Angeles, CA in 1996 and 2000, respectively. Between 2000 and 2002, he was a Postdoctoral Fellow at the Division of Engineering and Applied Science at the California Institute of Technology, Pasadena, CA. Currently, he is an Assistant Professor in the Department of Computer Science and Engineering and a faculty affiliate with the Digital Technology Center at the University of Minnesota. His research has focused on inertial navigation, sensing and estimation for distributed autonomous systems, sensor networks, and fault detection and identification. He has authored or coauthored more than 30 journal and conference papers in the above areas. He was the recipient of the Myronis fellowship (USC, 1998-2000). Finally, he has received grants from the Jet Propulsion Laboratory (JPL) and the National Science Foundation (NSF) for work pertinent to autonomous vehicle state estimation.*

Daniel Clouse *received his B.A. degree in Computer Science from the University of California, Berkeley in 1982, his M.S. degree in Computer Science from the University of California, San Diego (UCSD) in 1992, and his Ph.D. degree in Cognitive Science and Computer Science from UCSD in 1998. He is a Senior Member of the Technical Staff at the Jet Propulsion Laboratory, California Institute of Technology in the Machine Vision group of the Mobility Section. His interests include computer vision, machine learning, language translation, and word sense disambiguation.*

Larry Matthies *obtained a Ph.D. degree in Computer Science from Carnegie Mellon University in 1989, then moved to the Jet Propulsion Laboratory, where he is currently a Principal Member of Technical Staff and supervisor of the Machine Vision Group. His research has focused on terrain sensing and obstacle avoidance algorithms for autonomous navigation of robotic vehicles. At JPL, he pioneered the development of real-time algorithms for stereo vision-based obstacle detection and he contributed to the development of the structured light sensor used by the Sojourner Mars rover. He has also developed algorithms for visual motion estimation from image sequences, 3-D scene reconstruction from image sequences, real-time terrain classification using multispectral imagers, and environmental mapping using sonar and stereo vision sensors. His group currently has research projects on computer vision for robotic vehicles sponsored by NASA, DARPA, and the U.S. Army; these projects include work on navigation of Mars rovers, asteroid and comet landers, and Earth-based robotic vehicles for urban and cross-country missions. He is a member of the editorial board of the Autonomous Robots journal and an adjunct member of the Computer Science Department at the University of Southern California. He has also been an invited speaker at the Frontiers of Engineering Symposium organized by the National Academy of Engineering.*