

Design and Testing of the TeamIndus ECA Lunar Rover Navigation and Control

Venkat Bhargav*, Chhivilata S, J Yogeshwaran, Deepana G, Vishesh V, Rinku W, Midhun M, Karthic B
Guidance, Navigation and Control Engineer
TeamIndus - Axiom Research Labs Pvt. Ltd., Bengaluru, India

P. Natarajan

Senior expert
Guidance, Navigation and Control
TeamIndus - Axiom Research Labs Pvt. Ltd., Bengaluru, India

ABSTRACT

TeamIndus' Lunar Logistics vision includes multiple lunar missions over the coming years to meet requirements of science, commercial and efforts towards building readiness for crewed missions to Mars in the global exploration roadmap. TeamIndus is the only Indian team that participated in the Google Lunar X Prize. The challenge called for privately funded spaceflight teams to be the first to land a robotic spacecraft on the Moon, travel 500 meters, and transmit back high-definition video and images. The first mission is expected to have a net landed payload capacity of 50 kg. The prime objective is to demonstrate autonomous lunar landing, and operations for a Surface Exploration Rover - to collect data in the vicinity of the landing site. The rover is designed to execute the commands given by the operations manager autonomously. The rover software architecture relevant to the navigation and control is described in detail. The functional modes are defined to functionally distinguish the rover drive. Among the various operations performed by the rover are execution of simple drive steps through navigation and control algorithms. The kinematic model of the rover is studied in the hardware locomotion tests performed. The Kalman filter formulation to estimate the sensor bias and to get attitude estimate is discussed. The structural model and the frame definitions are described in a relevant section. A simple heading control algorithm is designed to control the heading direction of the rover according to the operations requirement. The path planning process and visual odometry processes are described. Finally the SIMULINK[®] model is tested on the prototype rover and the test experience results are discussed in the last section.

1 Introduction

With the current state of technology available today, it is highly dangerous for humans to explore other planetary surfaces because of radiation, unfavourable temperatures and unknown climate and environment conditions. This gives the motivation to use surface exploration rovers for continuing planetary exploration and extending the boundaries of humanity.

The planetary rovers are required to travel on rough, unexplored and uneven terrains to accomplish their scientific objectives without being trapped and tipping over [1] [2] [3]. Thus the planetary rovers are generally designed with articulated suspension systems in order to maximize traction distribution on all wheels and to improve tip-over stability limits [4]. The author Iagnemma et.al discusses methods to estimate wheel terrain contact angles and rough terrain control methods to maximize the traction for rovers.

The early lunar rovers include Lunokhod which was an eight wheeled rover with a mass of 900kg. Each wheel was driven by an independent motor and the rover is steered by differential rotation of wheels. Lunokhod was remotely operated

*Address all correspondence related to the paper to this author at venkat.bhargav28@gmail.com

by a five member team based on the images received during the traverse. While the contemporary lunar rover include Yutu rover which is based on 6-wheel independent drive with 4-wheel independent steering mechanism.

2 Teamindus ECA Rover Design

Teamindus lunar rover is chosen to land in the Mare Imbrium basin (29.52 deg N, 25.68 deg W), near Annegrit Crater, which is a regolith plain. Its main objective is exploring the moon surface and transmit HD video, images. In-order to minimize the cost of the mission, the mobility design of the rover is greatly simplified. The mass of the rover is restricted to approx 6kg which falls in to the category of the four wheeled differentially driven micro planetary rovers. Because of the mass constraint and simplification in the mobility design, the traversable terrain slope around the landing site region is chosen to be not more than 5°.

2.1 The Rover design is equipped with the following capabilities:

1. The rover shall capable of executing two types of drive modes
 - a Straight drives: The straight drives shall be constant velocity drives executed using time based cut-off and the heading errors of rover are corrected using gyro in closed loop.
 - b Point turns: The point turn drives shall be terminated onboard by using a time based cutoff based on targeted heading.
2. The rover shall be capable of driving in both forward and reverse directions.
3. The rover shall have capability to negotiate slopes on lunar regolith in the limits of $\pm 18^\circ$.
4. The rover shall have a pitch forward static stability limit of $\pm 55^\circ$, pitch backward static stability limit of $\pm 57^\circ$, roll left static stability limit of $\pm 60^\circ$, roll right static stability limit of $\pm 59^\circ$ on hard ground.
5. The rover has a single passive rocker arm suspension to the right of the chassis whose angle is constrained to a range of $\pm 17^\circ$ from nominal position. While the left wheels are independently fixed to the chassis.
6. The max velocity of the rover shall be 5 cm/s approx.
7. The ground clearance of the rover chassis is 0.1m including an expected worst case sinkage of 0.015m. Hence, any rocks or craters of size greater than 0.1m is considered as obstacle.
8. The rover shall compute and give the yaw, roll and pitch angles of its body with respect to the Terrain fixed frame using the rotation rates provided by the Inertial Measurement Unit(IMU) on-board. The orientation of rover body frame with respect to the absolute lunar ENU frame will be computed using the accelerometer and sun-sensor.
9. The rover shall estimate the position at the end of each drive using Visual Odometry (VO).

3 Overall Functional Architecture

The Figure.1 shows the functional modules in the architecture:

Guidance :

Local Path Planning: Performed using stereo image processing to identify waypoints within the resolvable range of the MastCam. The path is resolved into a series of Spot-Turns and Straight Drives.

Navigation:

Visual Odometry (VO): This is a localization process that utilizes stereo images obtained from the NavCams before & after a Drive step. VO provides both translation & rotation in the body frame.

Attitude Estimation (accelerometer + sun sensor): Roll and Pitch are obtained from accelerometer, with the Rover stationary between 2 drive steps. Sun sensor provides Heading angle. Attitude and gyro bias are estimated with respect to NED by using gravity vector and sun vector as measurement.

Attitude Propagation (gyros): Using the attitude estimated as initial condition, Rover attitude is propagated using gyros. This is performed during a Drive step.

Heading Control :

Straight Drive: Drive time is commanded (taking desired linear speed as a parameter & distance to next waypoint) + margin (to account for longitudinal slip). Heading is controlled throughout Straight Drive.

Spot-turn: Desired heading angle is commanded. This is performed with gyro-in-loop

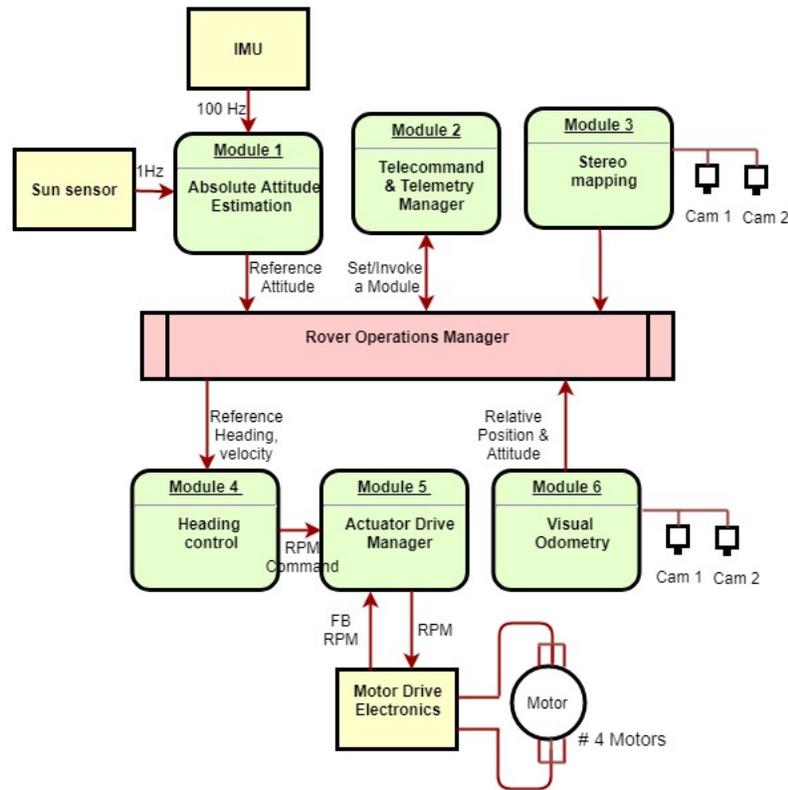


Fig. 1. Overall Functional Architecture

4 Local Path Planning

Local path planning (Path Guidance) is used to formulate next drive step for the Rover using stereo images. Before each drive step, a stereo image pair is acquired. Using the images, a safe spot is chosen and the rover path is calculated using stereo image processing in order to reach that safe spot.

The following steps explain the process involved in calculating the distance and the orientation in which, the rover has to move in order to reach the chosen safe spot.

4.1 Pre-processing

1. Read Stereo Images
2. Rectify the images using stereo calibration parameters.

4.2 User chooses the safe spot (2D Point from left camera image)

Local path planning uses this 2D point as input to calculate the orientation and the distance to which Rover has to drive to reach this safe spot. As shown in Figure 3, the 2D point (x, y) chosen by user is seen in left image as (x^l, y^l) and in right image as (x^r, y^r) and the corresponding 3D point is represented as $P(X, Y, Z)$. The method used to calculate the distance and orientation to get to point P is mentioned in the following steps.

1. When a 2D point is picked in the left image, a window is created around it and features within the window are matched in stereo pair.
2. Once the points are matched, the average disparity of the corresponding points in the window is obtained.
3. X and Z is calculated based on stereo geometry as follows.

$$Z = \frac{b \times f}{d} \quad (1)$$

$$X = \frac{x^l \times Z}{f} \quad (2)$$

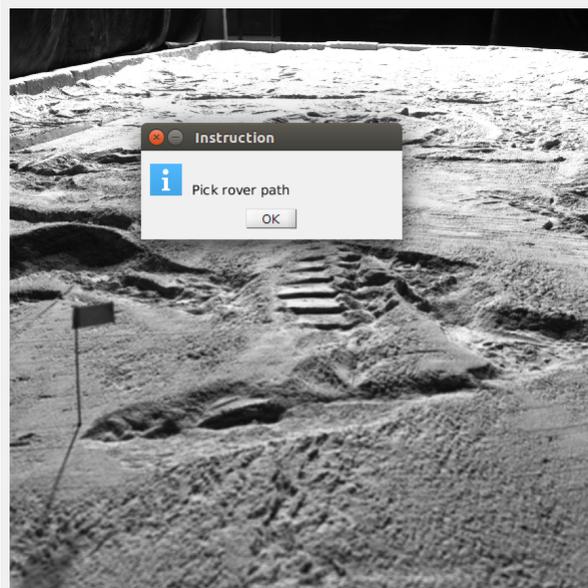


Fig. 2. User Interface to choose a safe spot

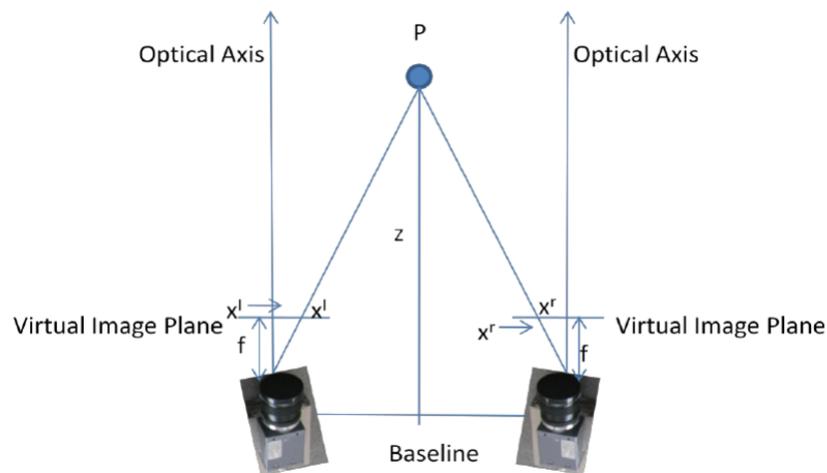


Fig. 3. Stereo Camera Geometry

where X, Z are the coordinates of 3D point P , b is baseline (0.16m) and f is focal length (12.5e-3m) and pixel pitch is 5.5e-6m.

- The distance to which Rover has to move to reach point $P(X,Z)$ from camera barycenter, $C(-b/2,0)$ is calculated as,

$$\text{straight drive distance} = \text{norm}(P,C) \quad (3)$$

- The orientation in which Rover has to move is calculated as,

$$\text{point turn angle} = \text{atan2}\left(\frac{(X - \frac{b}{2})}{Z}\right) \quad (4)$$

5 Visual Odometry

This is used to estimate the Rover location after each drive step using the current stereo image pair and the previous stereo image pair. Hence, Visual Odometry (VO) technique is exploited here in order to determine the relative change in the direction and magnitude of the movement of the Rover after each drive. The rotation and translation obtained is then used to update the position and attitude of the rover.

VO technique is adapted from an open source library libviso2[13] with stereo vision and is modified to be compatible for the lunar terrain. The 6 DOF motion of a stereo camera is computed based on minimizing the reprojection error of sparse feature matches. The following are the steps involved to estimate the egomotion of a stereo camera.

1. Input the internal and external camera calibration parameters.
2. Image Rectification
3. Extract SURF features and match features between consecutive stereo image pairs.
4. Pixel coordinates of features which survived the cycle match between $I_{left}^{current}$, $I_{right}^{current}$, $I_{left}^{previous}$ and $I_{right}^{previous}$ are extracted. Refer Figure 4.

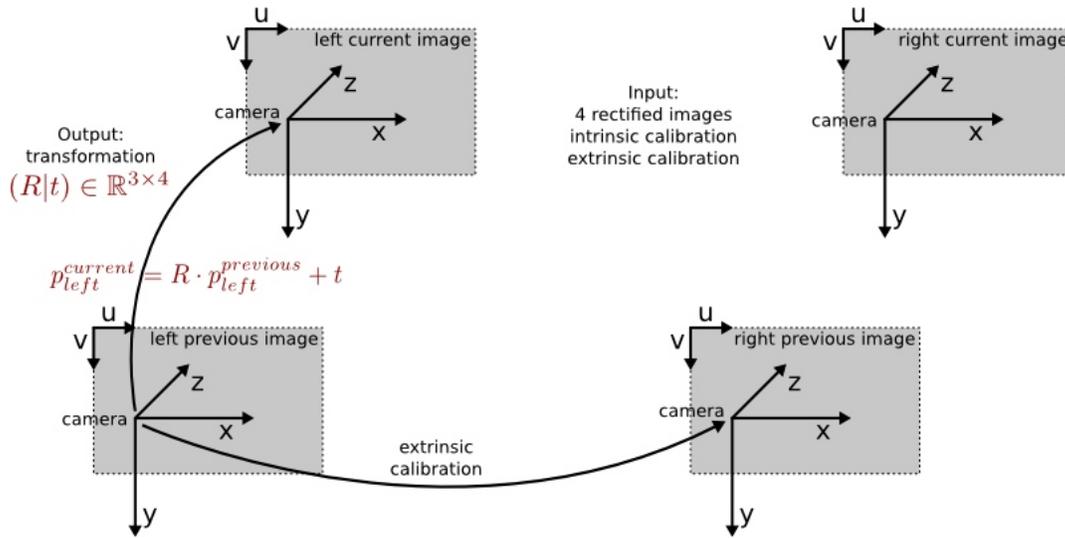


Fig. 4. VO Camera Geometry

5. Compute linear translation and rotation for the current image pair by transforming the survived pixel coordinates to world coordinates.
6. Compute cumulative pose.

For a better VO accuracy, a minimum of 50% percent overlap of features between the current and previous images must be maintained. With this constraint, the maximum straight drive distance is calculated to be 0.83 m and maximum turn angle to be 24 deg. As shown in Figure 5, the maximum straight drive distance is calculated as follows where $h = 0.33\text{m}$, $t = 15\text{deg}$.

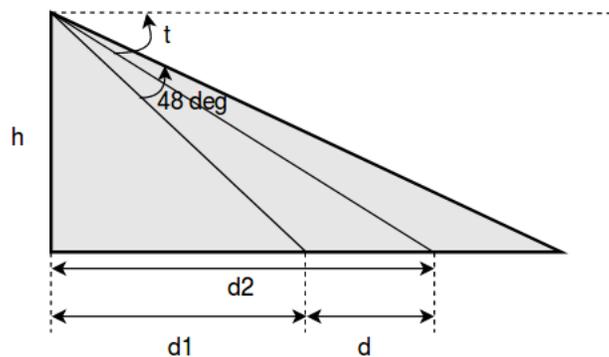


Fig. 5. Rover Maximum Straight drive and Turn Angle Calculation

$$d2 = h \times \tan(90 - t) = 1.23158 \quad (5)$$

$$d1 = h \times \tan(90 - 24 - t) = 0.40384 \quad (6)$$

$$d = d2 - d1 = 0.82774m \quad (7)$$

The maximum turn angle is calculated as $FOV/2 = 24deg$.

6 Simulation: Kinematic Modelling

The rover has a single passive rocker arm suspension which is pivoted at the centre right of the chassis. The right wheels are connected to the rocker arm while the left wheels are independently connected to the chassis. Since there is no steering mechanism for the wheels, the rover is steered using differential rotation of left and right wheels.

The kinematic model of the skid steering rover based on geometric approach [5] [6] is used for our modelling. The geometric approach assumes the flat terrain i.e. $Z=0$ all the wheels are always in contact with the ground. As shown in figure.6, the centre of the body frame x, y is located at the geometric centre of the rover. Let ψ denotes the yaw angle of the body frame with respect to the inertial frame, v_l, v_r denote the velocity of the left wheels and the velocity of the right wheels, W_t represent the wheel track. v_x, v_y denote the longitudinal and lateral velocity of the body frame x - y .

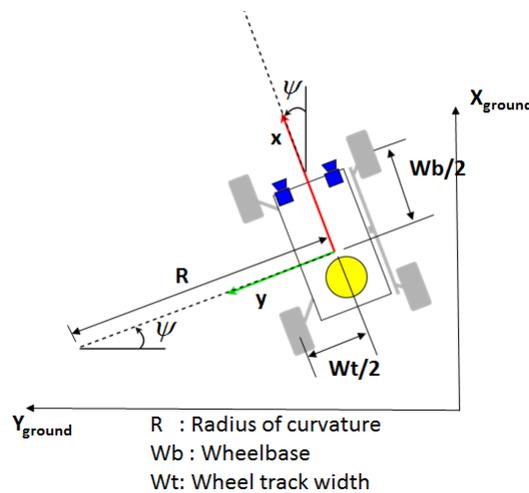


Fig. 6. Schematic diagram of the skid steered rover

$$v_x = \frac{v_r + v_l}{2} \quad (8)$$

$$v_y = 0 \quad (9)$$

$$\dot{\psi} = \frac{v_r - v_l}{W_t} \quad (10)$$

The skid steering rovers have lateral velocity component in the body frame. Let (x_{ICR}, y_{ICR}) be the coordinates of the instantaneous center of rotation in the body frame of the rover.

$$S = x_{ICR} = -\frac{v_y}{\dot{\psi}} \quad (11)$$

where v_y is the lateral velocity in the body frame and S is a constant called skid. Transforming the body velocity coordinates in to inertial frame coordinates we get

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \dot{\Psi} \end{bmatrix} \quad (12)$$

7 Simulation: Sensor Modelling

7.1 Gyro Model

The gyro measures angular rates or angular velocity in three axes, These rates are integrated to compute the change in orientation. The measured angular rates ω_{meas} include body rates, biases and noise quantity.

$$\omega_m = \omega_{true} + b + \eta_u \quad (13)$$

$$\dot{b} = \eta_v \quad (14)$$

Where ω_{true} is the true body angular rates in all three axis, rad/sec. b is the bias drift of the gyro. The stochastic noise parameters η_u, η_v are independent multivariate white Gaussian noise distribution with $N(0, \sigma_u^2), N(0, \sigma_v^2)$. σ_u is the angular random walk and σ_v is the rate noise density and

7.2 Accelerometer Model

The accelerometers measure the inertial acceleration, also called specific force to distinguish it from gravitational acceleration which on integration provides the vehicle speed. The measured acceleration from accelerometer is given by

$$A_b = A_i + \omega_b \times (\omega_b \times d) + \dot{\omega}_b \times d - \bar{g} \quad (15)$$

where A_i is the accelerations along the body axis. ω_b is body fixed body rates, $\dot{\omega}_b$ is body fixed angular accelerations, \bar{g} is gravity in body frame. d is distance vector of IMU with respect to CG. The measurement from accelerometer with error sources (i.e. bias and noise terms) is given by

$$A_{meas} = A_b + A_{bias} + \eta_a \quad (16)$$

where A_{bias} is accelerometer bias. η_a is the zero mean white Gaussian noise.

8 Simulation: Attitude estimation using EKF Design:

In order to execute drives and point-turns, heading knowledge is required at all times. Integrating the gyro rates from sensor measurements would drift the attitude, away from the true attitude with time. In order to correct for attitude drift, a filter has been designed which uses the roll and pitch information obtained from the accelerometers [7], [8] and yaw from sun sensor. The gravity vector is obtained by averaging the accelerometer measurements for a sec and used as an update for the filter. The filter estimates the biases of the gyro when the rover is stationary [9], [10], [11] and these biases are subtracted before executing any drive activity (i.e. point turn or straight drive). When driving (i.e. straight drive and point-turn), the heading is propagated using the gyros.

Given the gyro measurement ω_m the general attitude kinematics is given by:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} \omega_{m_x} \\ \omega_{m_y} \\ \omega_{m_z} \end{bmatrix} \quad (17)$$

$$(18)$$

Incorporating the stochastic model of the sensor in to the attitude kinematics, and rewriting the state equations by defining the state vector as $X = [\phi \ \theta \ \psi \ b_\phi \ b_\theta \ b_\psi]^T$ where ϕ, θ, ψ are the roll, pitch and yaw angles and b_ϕ, b_θ, b_ψ are the gyro biases in roll, pitch and yaw respectively.

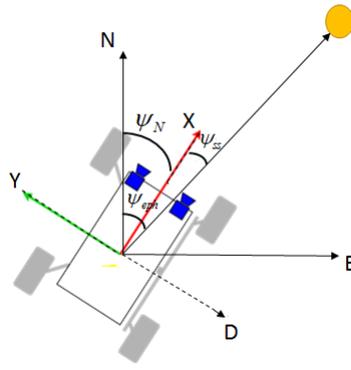


Fig. 7. Finding the rover heading w.r.t North using the sun's position from ephemeris data and sun angle w.r.t the body x-axis

$$\dot{X} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_{m_x} - b_\phi - \eta_{u_1} \\ \omega_{m_y} - b_\theta - \eta_{u_2} \\ \omega_{m_z} - b_\psi - \eta_{u_3} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \eta_{v_1} \\ \eta_{v_2} \\ \eta_{v_3} \end{bmatrix} \quad (19)$$

Since the above state equation is in non-linear form, so the non-linear Gauss-Markov process is given by

$$\dot{X}(t) = f(t, X(t), u) + G(t)W(t), \quad W(t) \sim N(0, Q(t)) \quad (20)$$

Simulation: Finding the Rover Heading w.r.t North

In the figure below ψ_N describes the rover heading angle w.r.t North in NED frame which is obtained by

$$\psi_N = \begin{cases} \psi_{eph} - \psi_{ss} & \text{if } (\psi_{eph} > \psi_{ss}) \\ \psi_{ss} - \psi_{ephe} & \text{otherwise} \end{cases} \quad (21)$$

where ψ_{eph} is the azimuth angle of the sun vector obtained from the ephemeris represented in NED frame. ψ_{ss} is the azimuth angle of the sun vector w.r.t rover body represented in NED frame.

The measurement model describes the gravity vector and sun vector in the NED frame. The measurement model is given by:

$$Z_k = h(X_k) + v_k = \begin{bmatrix} a_x \\ a_y \\ a_z \\ \psi \end{bmatrix} = \begin{bmatrix} -g \sin\theta \\ g \cos\theta \sin\phi \\ g \cos\theta \cos\phi \\ \psi \end{bmatrix}, \quad v_k \sim R_k$$

Where, a is the acceleration measured and ψ is the inertial yaw measurement, which is obtained from the sun sensor.

We use continuous-discrete extended kalman filter as suggested by Farrell [7] to write the state propagation and update equations. Now,

Propagation

$$F(X(t), t) \equiv \frac{\partial f}{\partial X} \Big|_{\hat{X}(t)} \quad (22)$$

$$\dot{\hat{X}}(t) = f(\hat{X}(t), u(t), t) \quad (23)$$

$$\dot{P}(t) = F(\hat{X}(t), t)P(t) + P(t)F^T(\hat{X}(t), t) + G(t)Q(t)G^T(t) \quad (24)$$

Gain

$$K_k = P_k^- H_k^T (\hat{X}_k^-) [H_k (\hat{X}_k^-) P_k^- H_k^T (\hat{X}_k^-) + R_k]^{-1} \quad (25)$$

$$H_k (\hat{X}_k^-) \equiv \left. \frac{\partial h}{\partial X} \right|_{\hat{X}_k^-} \quad (26)$$

Update

$$\hat{X}_k^+ = \hat{X}_k^- + K_k [\tilde{y}_k - h(\hat{X}_k^-)] \quad (27)$$

$$P_k^+ = [I - K_k H_k (\hat{X}_k^-)] P_k^- \quad (28)$$

9 Simulation: Control Design

The trajectory tracking or path following control needs the position estimate also as feedback to correct and follow the desired trajectory or path. Since maximum rover velocity is 5cm/s, the accelerometer measurements mostly include noise and cannot be used for position estimation. Also due to the lack of proper wheel odometry model for the skid steering rovers, the position estimate can not be obtained. Due to the above reasons way points targeting is not implemented, so a simple heading/steering control is considered for executing drives. The heading control nullifies any disturbance heading error set during the drives and it drive the rover in the commanded heading direction.

The steering control for the rover is achieved by maintaining the difference in rotations per minute (rpm) between the left side wheels and right side wheels. The left side wheels are given same command constantly, similarly the right side wheels, so that there will not be any rpm difference between any two wheels on the same side. There will be inner control loop (rpm control) for each motor to maintain the commanded rpm. Even though each motor has variations in characteristics, it is assumed that the inner control loop (rpm control) will try to compensate for the errors.

The control algorithm formulation is based on the kinematic model of the rover. The kinematic equations of motion 10 include the heading rate of the rover. The heading rate is a linear model with type 1 system characteristics. A PD control algorithm is chosen to control the heading of the rover.

The architecture of the simulation flow is shown in figure 8. The outer control loop is a heading control, which is a simple PD controller. The heading angle is taken from the gyro sensor to get the heading angle. The inner loop includes the motor loop with a PI controller and the motor transfer function.

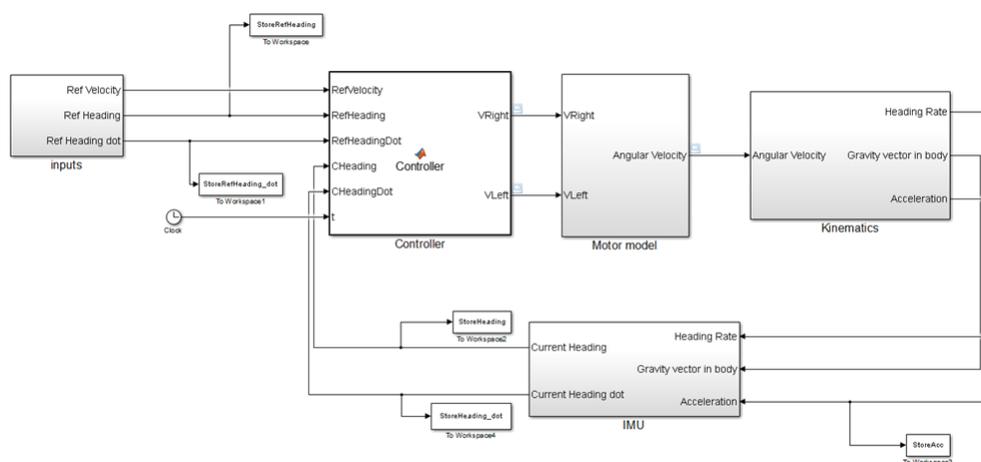


Fig. 8. The Simulation flow block diagram

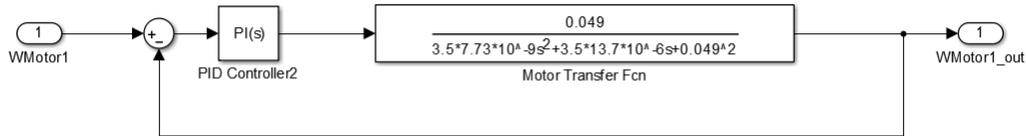


Fig. 9. The Motor inner control loop

Let ψ_d , $\dot{\psi}_d$ is the desired heading angle and desired heading rate to be achieved. V_{ref} is the reference velocity along the heading direction. The control commands for the rover is calculated as follows:

$$\psi_{er} = kp(\psi_d - \psi) + kd(\dot{\psi}_d - \dot{\psi}) \quad (29)$$

$$v_r = V_{ref} + \psi_{er} \cdot w/2 \quad (30)$$

$$v_l = V_{ref} - \psi_{er} \cdot w/2 \quad (31)$$

10 Hardware

The rover design is shown in the figure 10 and front, back view is shown in figure 12. The design parameters of the rover configuration is given in the table 1.

Table 1. Key design parameters

SI No.	Parameter	Value
1	Wheel diameter	100 mm
2	Wheel width	75 mm
3	Wheel Base	450 mm
4	Wheel track	443 mm
5	Ground clearance	115 mm
6	Camera height	238 mm
7	Chassis length	312 mm
8	Chassis width	177 mm
9	Chassis height	106 mm
10	Nominal linear speed	5 cm/s
11	Steering	Skid/differential
12	Obstacle size	100 mm

10.1 Sensors

The IMU location with respect to the Body Frame is shown in figure 11. The IMU characteristics [12] is provided in the table 2. There are two stereo cameras looking in the front direction of the rover and are fixed to the chassis by a tilt angle of 20° . These are used for Visual Odometry and stereo mapping. Visual Odometry is used to update the attitude and position of the rover at the end of the drive. There is one Hazard camera looking to the rear of rover. Each wheel is driven by a individual motor and each motor is equipped with hall sensors which senses the position of the rotor. The motor driver card process this and gives the rpm (rotations per minute) feedback from the motor.

10.2 Actuators

The four motors are independently controlled to rotate the four wheels of the rover. Each motor is driven by an independent motor driver card. Each motor is also fixed with a gear box ratio of 271:1, in order to maximise the torque provided by

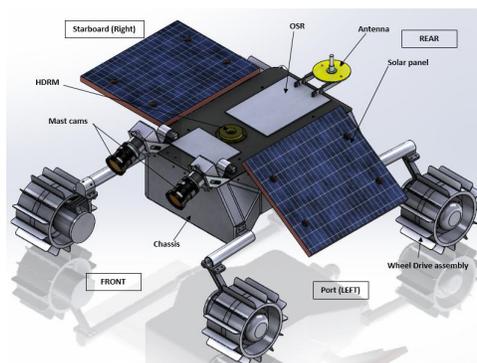


Fig. 10. The over view of the rover

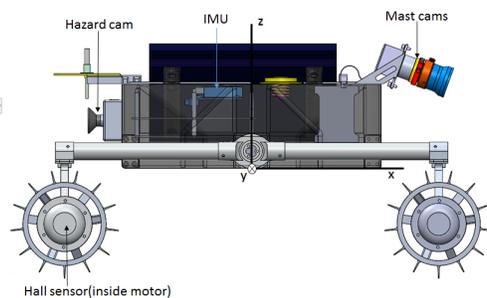


Fig. 11. Figure showing the various sensors w.r.t body frame

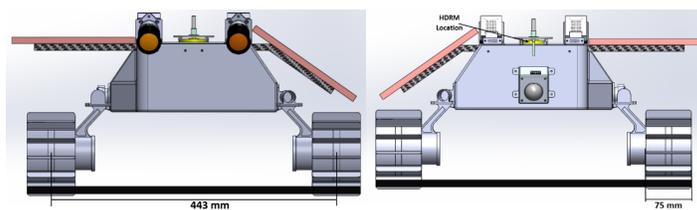


Fig. 12. The front view and back view of the rover

Table 2. IMU Specification table

Gyroscope Parameter	comments	Value	Unit
Bias Repeatability	$-40^{\circ} \text{ C} \leq T_A \leq +85^{\circ} \text{ C}$, 1σ	± 0.2	$^{\circ}/\text{sec}$
In-Run Bias Stability	1σ	6.25	$^{\circ}/\text{hr}$
Angular Random Walk	1σ	0.3	$^{\circ}/\sqrt{\text{hr}}$
Rate Noise Density	$f = 25$, no filtering	0.0066	$^{\circ}/\text{sec}/\sqrt{\text{hr}} \text{ rms}$
Accelerometer Parameter	comments	Value	Unit
Bias Repeatability	$-40^{\circ} \text{ C} \leq T_A \leq +85^{\circ} \text{ C}$, 1σ	± 3	mg
In-Run Bias Stability	1σ	32	μg
Velocity Random Walk	1σ	0.023	$\text{m}/\text{sec}/\sqrt{\text{hr}}$
Noise Density	$f = 25$, no filtering	0.055	$\text{mg}/\sqrt{\text{HZ}} \text{ rms}$

the motor.

11 Tests and Results

11.1 Attitude Estimation Simulation

The EKF algorithm is simulated considering initial state vector as $X = [0 \ 0 \ 0 \ 0 \ 0]^T$ and initial state error covariance matrix $P = \text{diag}(0, 0, 0, 0.01, 0.01, 0.01)$. The figure 14 shows how the estimate of bias model is able to follow the sensor bias model (as discussed in section under gyro sensor model). The figure 13 shows the attitude estimation accuracy of the EKF. The estimation accuracy on roll and pitch is approximately 0.2 deg, while on yaw it is less than 0.1 deg. The process noise covariance, measurement noise covariance is chosen based on the sensor specifications of angular random walk and rate noise density, velocity random walk given in the table 2.

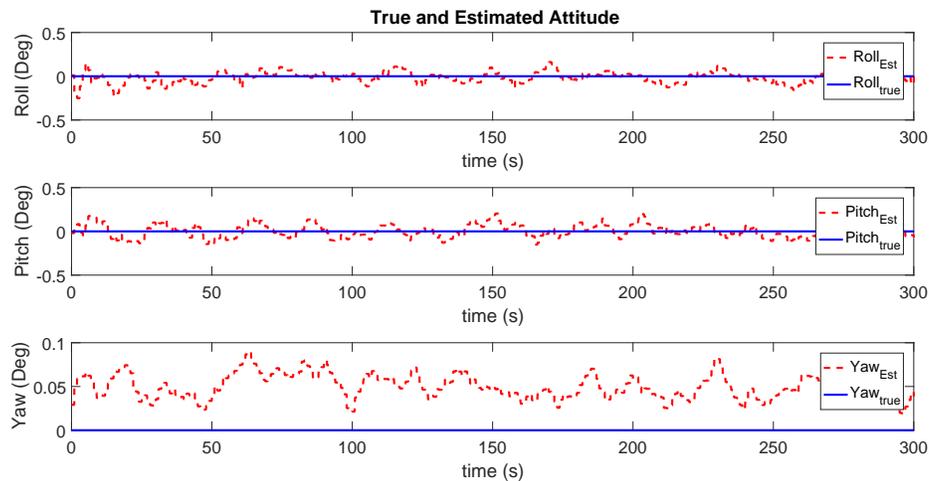


Fig. 13. Attitude Estimate Plot

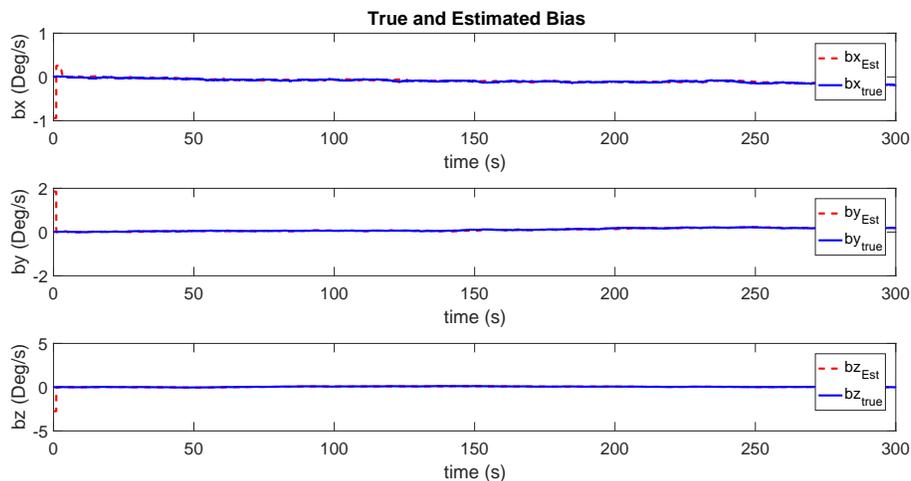


Fig. 14. Bias Estimate plot

The control algorithm discussed in the previous section is simulated along with the entire modelling of the kinematics, sensor models and estimation algorithms. The heading control just tries to compensate for any heading errors occurred by following the desired angle. The figure 15 shows the heading control tracks the desired heading angle.

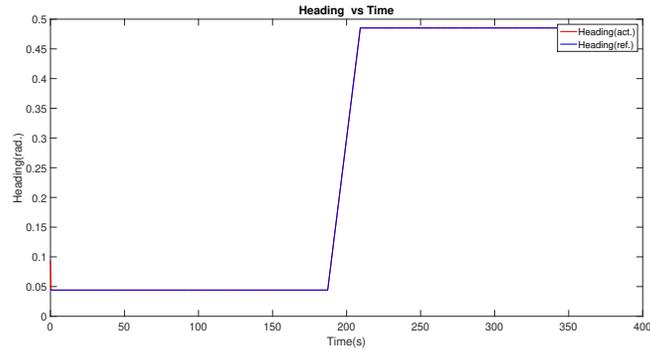


Fig. 15. Tracking the reference heading profile

11.2 Drive on Prototype Rover

In order to check the initial operations performance on the prototype rover a loose sand test bed is created. The test bed is a 10m by 10m sand pit created at the test facility. The prototype model is able to perform the straight drive and point turn operations as per the simple kinematic model given in equations 8-10. The figure 16 shows the track of the rover wheels during straight drive. The figure 17 shows the point turn operation performed by the rover with differential velocity on left and right wheels.

The entire modelling and algorithm design is done in SIMULINK[®]. We use embedded code generation of MATLAB[®] for generating generic 'C' code to run on 'Linux' platform. This gave us a lot of flexibility and freedom to focus only on algorithm design.

When we ran generic code on the target hardware without estimation algorithm, the bias in the "IMU" drifts the rover away from the commanded heading direction. So before giving any drive command we first run the estimation algorithm to estimate the bias. This estimated bias is subtracted when ever heading control operations are performed. The rover can be commanded either straight drive or point turn. In point turn operation, the rover can be commanded to turn a desired heading angle with respect to the north. When it is tested on the prototype rover it turns to the desired heading angle with 0.5 deg accuracy. While for straight drive operations, the rover can be commanded a reference speed along with the heading angle which makes the rover to drive along the desired heading direction with the commanded speed. In this scenario, we have a problem of lateral offset when compared to the assumed profile path if any heading errors occur. This problem can be solving by designing a trajectory tracking controller with position feedback in the loop of estimation.

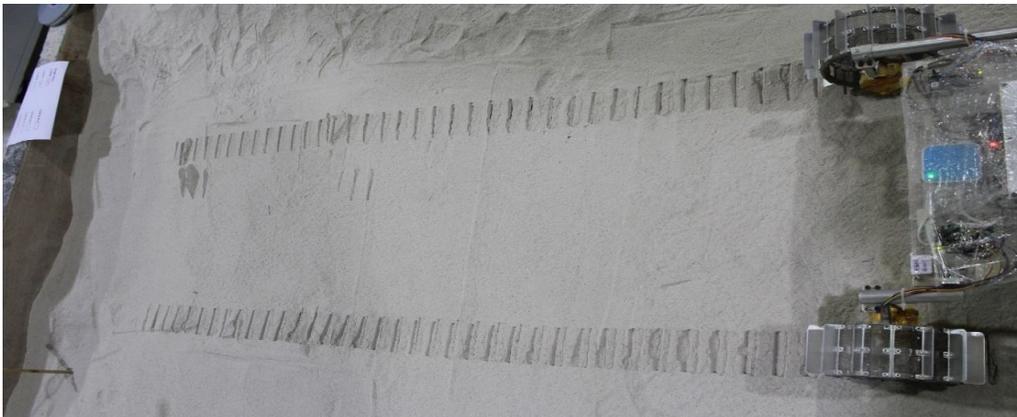


Fig. 16. Figure showing track of the wheels during straight drive

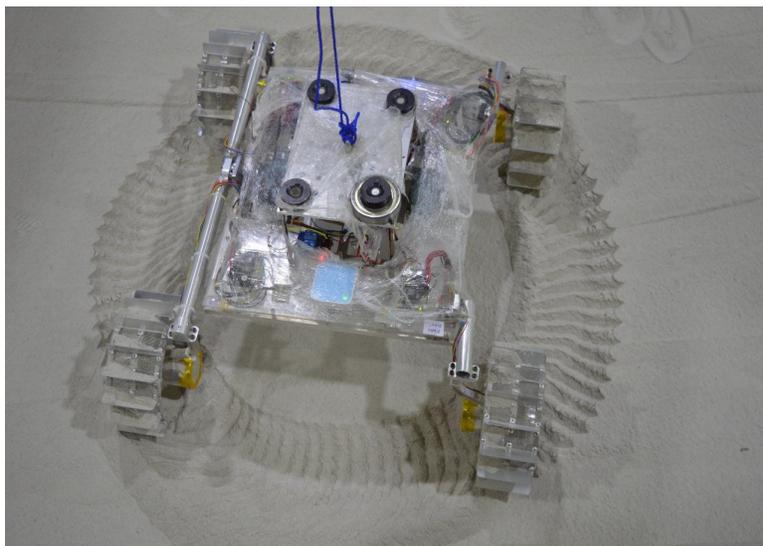


Fig. 17. Figure showing point turn operation by the rover

11.2.1 Single wheel jam

Some test case studies are done if a single wheel of the rover gets jammed due to some problem. Figure 18 shows the wheel track trace curve when front left wheel of the rover gets jammed. Figure 19 shows the trench formation at the wheel.

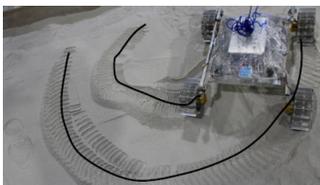


Fig. 18. Sample Case – 4.3RPM – FL Jam - Top view with wheel track trace curve



Fig. 19. Sample Case – 4.3RPM – FL Jam - Soil pile up near the torquer and trench formation

This contingency can be mitigated by the following actions:

1. Drive at lower RPM with the jammed wheel trailing behind.
2. Do not command the mirror side wheel, if traction is not an issue. The jammed wheels must be trailing behind.

11.3 Rover Path Guidance and Navigation using Local Path Planning and Visual Odometry

Path guidance test is performed by planting small flags on the sand test bed as markers. Safe spot is chosen at the root of the flag present in the left camera image as shown in Figure 20. Local path planning uses this 2D point as input to calculate the orientation and the distance to which Rover has to drive to reach this flag.

Once the Rover has moved based on the command given using local path planning, the next pair of image is acquired so that visual odometry can be performed. Figure 21 shows the tyre marks of Rover over the planted flags which serves as

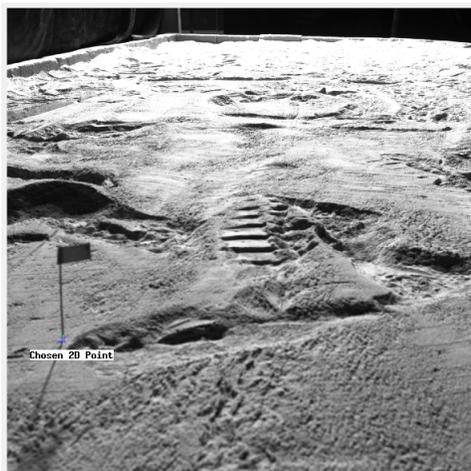


Fig. 20. User choosing the root of the flag as destination

the visual verification that the Rover targeted the flags as commanded. Figure 22 shows the comparison of Rover position at each drive step w.r.t Local path planning, Visual Odometry and the Ground truth. Here, ground truth is obtained using tape measurements and compass.



Fig. 21. Rover Tyre tracks validating Local Path Planning

Figure 23 shows the error plots of Visual Odometry compared to Ground truth and Figure 22 shows the error plots of Local Path Planning compared to Ground truth.

Method	Max Distance Error (cm)	Avg Distance Error (cm)	Max Orientation Error (deg)	Avg Orientation Error (deg)
Local Path Planning				
Visual Odometry	9	5.1	6.3	2.75

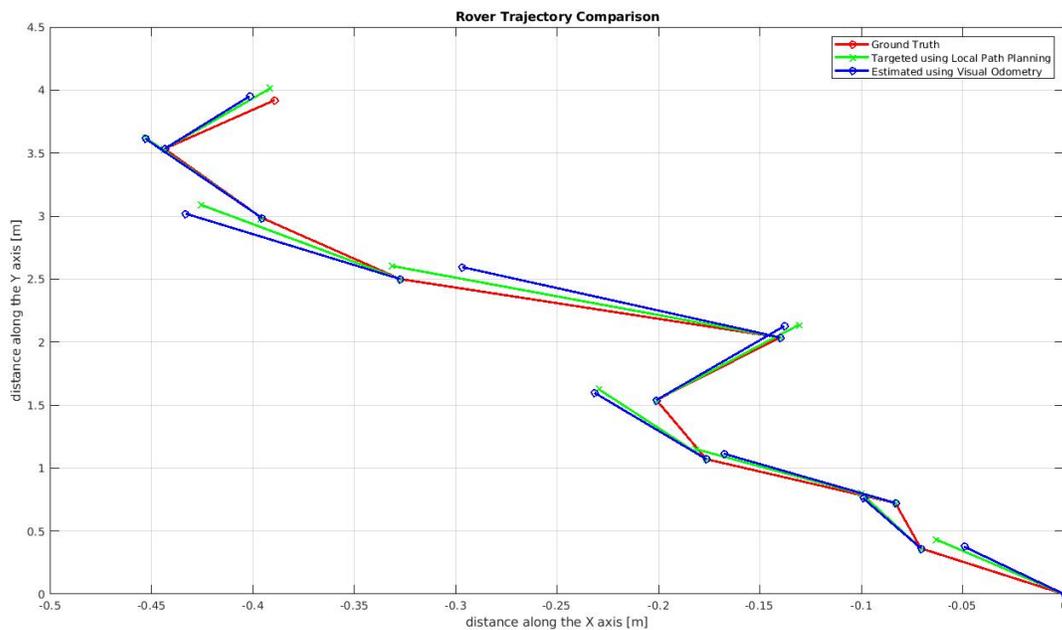


Fig. 22. Comparison of Rover trajectory

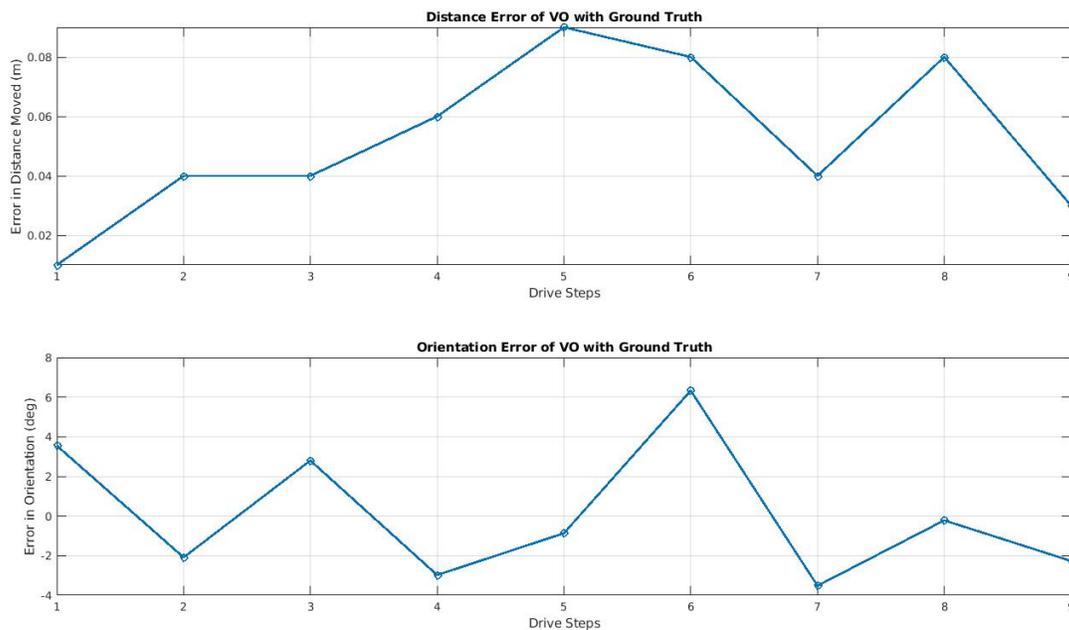


Fig. 23. Visual Odometry: Error Plots

12 Conclusions

The design, analysis, simulation and testing of the ECA lunar rover's guidance, navigation and control were shown in this paper. For a lightweight rover weighing around 6 kgs, the ECA has powerful capabilities to traverse the lunar terrain and achieve the objectives of nearside lunar mission. The functional architecture provides for a human-in-the-loop system with the rover software designed to perform absolute attitude estimation, received and generate telemetry and telecommands

respectively, perform stereo mapping through images received from rover cameras. Using the inputs from the plan, the rover onboard software allows to track the desired heading and speed throughout the motion towards intermediate waypoints. Visual Odometry is used to estimate relative displacement between the start and the stop of the drive using cameras. A drive-planning User Interface is built to allow operators to choose target positions visually. Modeling of the kinematics, sensors, control provides the base for a drive and attitude estimation EKF simulation. Once the simulation results seemed to meet drive requirements, the onboard software was ported to the rover controller hardware onboard the Prototype rover. A 10m x 10m sand pit was created to test-drive the rover through various scenarios. Drive observations and lessons learnt have been provided. The capability to drive up slopes up to 20 deg was demonstrated The Visual Odometry results have yielded average distance errors around 9 cm which meets the requirement. This summarizes an overview of the ECA lunar rover guidance, navigation and control system.

Acknowledgements

The rover development has been the result of the hardwork extended by multiple subsystems across TeamIndus. The structures team led by Karan Vaish and Paul were instrumental in the rover structures and systems. The motor control was developed by Hari. The Guidance, Navigation and Control team were responsible for the control system design and software.

References

- [1] Leger, P. C., Trebi-Ollennu, A., Wright, J. R., Maxwell, S. A., Bonitz, R. G., Biesiadecki, J. J., Hartman, F. R., Cooper, B. K., Baumgartner, E. T., and Maimone, M. W., 2005. "Mars exploration rover surface operations: Driving spirit at gusev crater". In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, Vol. 2, IEEE, pp. 1815–1822.
- [2] Biesiadecki, J. J., Baumgartner, E. T., Bonitz, R. G., Cooper, B., Hartman, F. R., Leger, P. C., Maimone, M. W., Maxwell, S. A., Trebi-Ollennu, A., Tunstel, E. W., et al., 2006. "Mars exploration rover surface operations: Driving opportunity at meridiani planum". *IEEE robotics & automation magazine*, **13**(2), pp. 63–71.
- [3] Erickson, J., Adler, M., Crisp, J., Mishkin, A., and Welch, R., 2002. "Mars exploration rover: surface operations".
- [4] Iagnemma, K., and Dubowsky, S., 2010. *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*. Springer Publishing Company, Incorporated.
- [5] Solc, F., and Sembera, J., 2008. "Kinetic model of a skid steered robot". In *Proceedings of the 7th WSEAS International Conference on Signal Processing, Robotics and Automation*. Cambridge, UK, pp. 61–65.
- [6] Pazderski, D., and Kozlowski, K., 2008. "Trajectory tracking of underactuated skid-steering robot". In *American Control Conference, 2008, IEEE*, pp. 3506–3511.
- [7] Farrell, J., 2008. *Aided navigation: GPS with high rate sensors, page no 357*. McGraw-Hill, Inc.
- [8] Fuke, Y., and Krotkov, E., 1996. "Dead reckoning for a lunar rover on uneven terrain". In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, Vol. 1, IEEE, pp. 411–416.
- [9] Ali, K. S., Vanelli, C. A., Biesiadecki, J. J., Maimone, M. W., Cheng, Y., San Martin, A. M., and Alexander, J. W., 2005. "Attitude and position estimation on the mars exploration rovers". In *2005 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, IEEE, pp. 20–27.
- [10] Baumgartner, E. T., Aghazarian, H., and Trebi-Ollennu, A., 2001. "Rover localization results for the fido rover". In *Intelligent Systems and Advanced Manufacturing, International Society for Optics and Photonics*, pp. 34–44.
- [11] Vaganay, J., Aldon, M.-J., and Fournier, A., 1993. "Mobile robot attitude estimation by fusion of inertial data". In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, IEEE, pp. 277–282.
- [12] ADIS16485 analogue devices - six degrees of freedom inertial sensor. <http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16485.pdf>. Accessed: 2018-03-11.