

# Body-fixed SLAM with Local Submaps for Planetary Rover

Bo Zheng<sup>1</sup>, Zexu Zhang<sup>1</sup>, Jing Wang<sup>2</sup>, Feng Chen<sup>3</sup> and Xiangquan Wei<sup>3</sup>

<sup>1</sup>(School of Astronautics, Harbin Institute of Technology, 150080 Harbin, China)

<sup>2</sup>(Science and Technology on Optical-Radiation Laboratory, 100000 Beijing, China)

<sup>3</sup>(Shanghai Institute of Aerospace System Engineering, 201108, Shanghai, China)

(E-mail: [zexuzhang@hit.edu.cn](mailto:zexuzhang@hit.edu.cn))

In traditional Simultaneous Localisation and Mapping (SLAM) algorithms based on Extended Kalman Filtering (EKF-SLAM), the uncertainty of state estimation will increase rapidly with the development of the exploration process and the increase of map area. Likewise, the computational complexity of the EKF-SLAM is proportional to the square of the number of feature points contained in the state variables in a single filtering process. A new SLAM algorithm combining the local submaps and the body-fixed coordinates of the rover is presented in this paper. The algorithm can reduce the computational complexity and enhance computational speed in consideration of the processing capability of the onboard computer. Due to the introduction of local submaps, the algorithm represented in this paper is able to reduce the number of feature points contained in the state variables in each single filtering process. Therefore, the algorithm could reduce the computational complexity and improve the computational speed. In addition, rover body-fixed SLAM could improve the navigation accuracy of a rover and decrease the cumulative linearization error by coordinates transformation during the update process, which is shown in the simulation results.

## KEY WORDS

1. Planetary Surface Exploration. 2. EKF-SLAM. 3. Local Submaps. 4. Body-fixed SLAM.

Submitted: 21 September 2018. Accepted: 27 May 2019. First published online: 26 June 2019.

1. INTRODUCTION. Deep space exploration has become a hot research topic for several countries, and the development of a rover system for complex environments has become a key issue in planetary surface exploration. The Simultaneous Localisation and Mapping (SLAM) algorithm can achieve self-localisation without prior information by constructing an incremental map, and this technique has great potential for rover systems (Islam et al., 2017).

In general, semi-autonomous positioning methods are common for rover surface exploration, but accuracy is affected by the reliability of the rover system and the computational ability of the onboard computer (Meng and Hutao, 2014). The methods are unstable and inaccurate when the rover drives on rough terrain, because the wheels of the rover may skid. In such cases, the rover can obtain relative position and altitude information more easily

through a SLAM algorithm, and this can play an important role in autonomous navigation missions (Tan et al., 2013).

The original SLAM algorithm was presented by Leonard and Durrant-Whyte (1991) and could locate position while modelling environmental maps. The Extended Kalman Filter (EKF)-SLAM algorithm was established by Dissanayake et al. (2001) and is based on a Kalman Filter (KF) on the general computational framework of a SLAM algorithm for mobile robots. In the EKF-SLAM algorithm, Dissanayake et al. (2001) proved that a solution to the SLAM problem is indeed possible and gave the underlying structure of the solution to the SLAM problem.

In early SLAM research, scholars focussed on a Two-Dimensional (2D) model of the SLAM problem, but this is not suitable for complex planetary terrain. Therefore, the Six-Degree Of Freedom (6-DOF) SLAM for outdoor environments was developed by Nuchter et al. (2007).

The traditional EKF-SLAM method has an obvious limitation: the computational complexity is proportional to the square of the number of feature points contained in the state variables in a single filtering process. Many scholars have focussed on reducing the computational complexity of EKF-SLAM. Martinez-Cantin and Castellanos (2006) addressed the consistency issue of the EKF-SLAM problem and proposed a local map sequencing algorithm, which can reduce the computational cost in the majority of updates. The submaps method was presented as a useful way to solve this problem by Paz and Neira (2006) and Joly and Rives (2009). The computational complexity will decrease by this method.

Also, the limitation of EKF linearization errors is a big problem for EKF-SLAM. The existing methods include the alternative linearization method (Martinez-Cantin and Castellanos, 2005) and nonparametric method (Thrun et al., 2004). The traditional EKF-SLAM method is established in global coordinates, then the accumulated uncertainty of the pose of the rover and the uncertainty of the position of feature points are introduced into the observation equation, which increases the linearization error. In the body-fixed SLAM method, the feature points will be updated in the body coordinates of the rover; only the uncertainty of camera range and Inertial Measurement Unit (IMU) measurement noise are added to the linearization of the observation equation, which greatly reduces the linearization error.

In addition, the terrain of the planet is complex, and the computing ability of the onboard computer is limited. It is necessary to reduce the computational complexity, increase the computational speed and decrease the algorithm error during the large-scale cruising exploration. A flow chart of this paper's structure is given in Figure 1.

2. MOTION MODEL OF THE ROVER. The set of the coordinate systems adopted to describe the rover's motion is shown in Figure 2.

2.1. *Notation and hypotheses.* In the following:  $\mathbf{O}_G - x_G y_G z_G$  is the global coordinate system, which is considered inertial.  $\mathbf{O}_R - x_R y_R z_R$  is the body coordinate system of the rover, where  $x_R$  points to the direction of the rover's motion and  $y_R$  points to the left direction of the rover's motion.  $\mathbf{O}_C - x_C y_C z_C$  is the coordinate system of the binocular camera, where  $x_C$  points to the vertical direction of the camera lens and  $z_C$  points to the opposite direction of the connecting stick.  $\mathbf{O}_{A_i} - x_{A_i} y_{A_i} z_{A_i}$  is the coordinate system of the rover's  $i$ -th wheel.  $\mathbf{O}_{W_i} - x_{W_i} y_{W_i} z_{W_i}$  is the coordinate system of the contact point for the rover's  $i$ -th

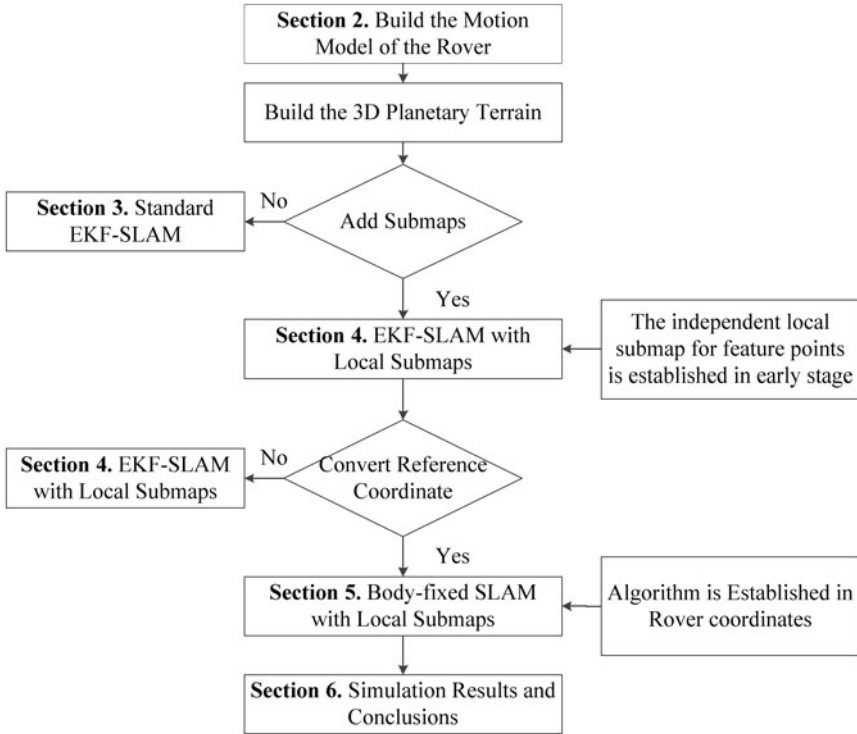


Figure 1. A flow chart of the paper's structure.

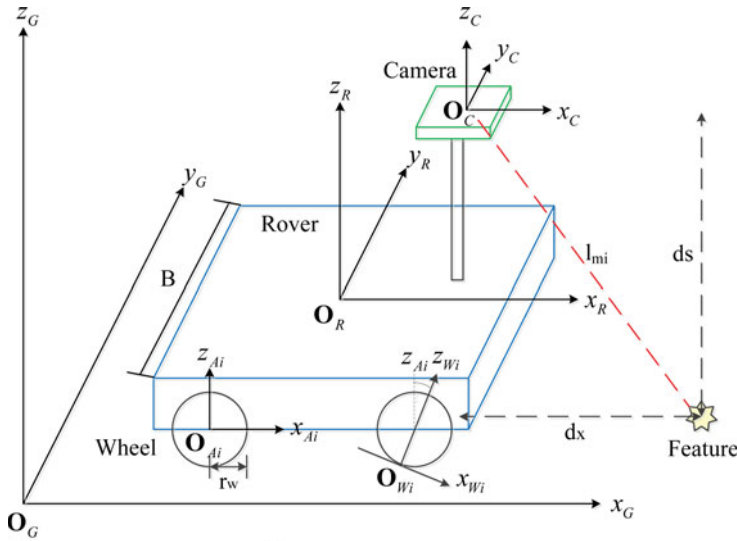


Figure 2. Coordinate system of the rover's motion.

wheel and the planet surface, where  $x_{W_i}$  points to the tangential direction of the contact point's surface and  $z_{W_i}$  points to the normal direction of the contact point's surface.  $l_{mi}$  is the distance between the  $i$ -th feature point and the binocular camera.  $d_x$  is the distance between the  $i$ -th feature point and the binocular camera along the  $x_C$  coordinate axis.  $d_s$  is the distance between the  $i$ -th feature point and the binocular camera along the  $z_C$  coordinate axis.  $[a, b, c]$  is the installation location of the binocular camera in the  $\mathbf{O}_R - x_R y_R z_R$  coordinate of the rover.  $r_w$  is the radius of the rover's wheel and  $B$  is the width of the rover.

2.2. *Kinematic model of the rover.* The rover investigated in this paper is a four-wheel differential rover with three translational Degrees Of Freedom (DOF) and three rotational DOF. The attitude of the rover is determined by the contact point of the soil surface and the rover. Linear velocity and angular velocity are control variables during the process of the rover's motion, which is determined by the angular velocity of two sets of differential wheels (Yen et al., 1999).

$[x_R, y_R, z_R, \phi_x, \phi_y, \phi_z]^T$  denotes position and attitude vectors of the rover, where  $\phi_z, \phi_y, \phi_x$  stand for yaw angle, pitch angle and roll angle, respectively. The motion model equation is the following:

$$\begin{aligned} \dot{x} &= r_w \left( \frac{\dot{\theta}_L(t) + \dot{\theta}_R(t)}{2} \right) \\ \omega_z &= \frac{r_w}{B} (\dot{\theta}_L(t) - \dot{\theta}_R(t)) \end{aligned} \tag{1}$$

where  $\dot{\theta}_L(t)$  is the rotation velocity of the left wheel,  $\dot{\theta}_R(t)$  is the rotation velocity of the right wheel,  $r_w$  is the radius of the wheel,  $B$  is the distance between two wheels.  $\mathbf{v}(t) \in \mathfrak{R}^3$  is the linear velocity which points to the  $x$  axis,  $\boldsymbol{\omega}(t) \in \mathfrak{R}^3$  is the angular velocity which points to the  $z$  axis,  $\dot{x}$  and  $\omega_z$  are parameters of linear velocity and angular velocity.  $\mathbf{v}(t) = [\dot{x}, 0, 0]^T$  and  $\boldsymbol{\omega}(t) = [0, 0, \omega_z]^T$ .

The control inputs are controlled in the body coordinate system of the rover. Therefore, the linear velocities and angular velocities need to be converted to a global coordinate system. The rotation matrix  $C_R^G$  is expressed as:

$$C_R^G = \begin{bmatrix} c\phi_y s\phi_z & -c\phi_x s\phi_z + s\phi_x s\phi_y s\phi_z & s\phi_x s\phi_z + c\phi_x s\phi_y c\phi_z \\ c\phi_y c\phi_z & c\phi_x c\phi_z + s\phi_x s\phi_y s\phi_z & -s\phi_x c\phi_z + c\phi_x s\phi_y s\phi_y s\phi_z \\ -s\phi_y & s\phi_x c\phi_y & c\phi_x c\phi_y \end{bmatrix} \tag{2}$$

where  $c$  represents cosine,  $s$  represents sine and  $C_R^G$  represents the transformation matrix from the rover coordinates to global coordinates.

The transformation matrix of the angular velocity  $S_\omega^G$  is:

$$S_\omega^G = \begin{bmatrix} 1 & s\phi_x t\phi_y & c\phi_x t\phi_y \\ 0 & c\phi_x & -s\phi_x \\ 0 & \frac{s\phi_x}{c\phi_y} & \frac{c\phi_x}{c\phi_y} \end{bmatrix} \tag{3}$$

where  $t$  represents tangent and  $S_\omega^G$  represents the transformation matrix of the angular velocity. The entire kinematic model equation of the rover can be written as:

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{z}_R \\ \dot{\phi}_x \\ \dot{\phi}_y \\ \dot{\phi}_z \end{bmatrix} = \begin{bmatrix} C_R^G & 0 \\ 0 & S_\omega^G \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4}$$

where  $v$  is the linear velocity of the rover and  $w$  is the angular velocity.

Since the attitude of the rover is determined by the contact point of the soil and the rover, the motion of a single wheel needs to be analysed. The coordinates of the contact point can be obtained in the coordinate system of the rover’s wheel:

$$\begin{bmatrix} X_{W_i}^{A_i} \\ Y_{W_i}^{A_i} \\ Z_{W_i}^{A_i} \\ 1 \end{bmatrix} = C_{W_i}^{A_i} \begin{bmatrix} X_{W_i} \\ Y_{W_i} \\ Z_{W_i} \\ 1 \end{bmatrix} \quad i = 1, 2, 3, 4 \tag{5}$$

where  $i$  represents the  $i$ -th wheel of the rover,  $C_{W_i}^{A_i}$  is the transformation matrix from the coordinate system of the contact point to the coordinate system of the rover and  $(X_{W_i}, Y_{W_i}, Z_{W_i})$  are the coordinates of the contact point.  $C_{W_i}^{A_i}$  can be represented as:

$$C_{W_i}^{A_i} = \left[ \begin{array}{ccc|c} \cos \delta_i & 0 & \sin \delta_i & -r_w \sin \delta_i \\ 0 & 1 & 0 & 0 \\ -\sin \delta_i & 0 & \cos \delta_i & -r_w \cos \delta_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \tag{6}$$

where  $\delta_i$  is the angle between the normal vector of the contact point and the  $z_{A_i}$  axis of coordinate system of the rover.

2.3. *Installation model of the binocular camera.* Three-dimensional ranging information is a common measurement in the localisation system of rovers. A binocular camera is used as a ranging sensor in this paper. The navigation observations are a series of three-dimensional point clouds which are generated by matching pairs of visual features. The installation model of the binocular camera on the rover is shown in Figure 3 (vertical view) and Figure 4 (lateral view).

The binocular camera is fixed on an oblique stick with two DOF at a certain angle. The stick, whose length is  $L_a$ , is able to rotate around the connecting bolt with an angular velocity  $\omega_\phi$ .  $\phi_L$  is the angle between the control stick and the vertical direction of the platform.  $\theta_L$  is the angle between the control stick and the horizontal forward direction of the rover. The coordinate system of the camera has its origin at the connecting bolt for convenience. The stick can also rotate with an angular velocity  $\omega_\delta$  about the platform system. A wide range of the three-dimensional terrains is achieved by the camera’s control inputs  $u_L = [\omega_\delta, \omega_\phi]$ .

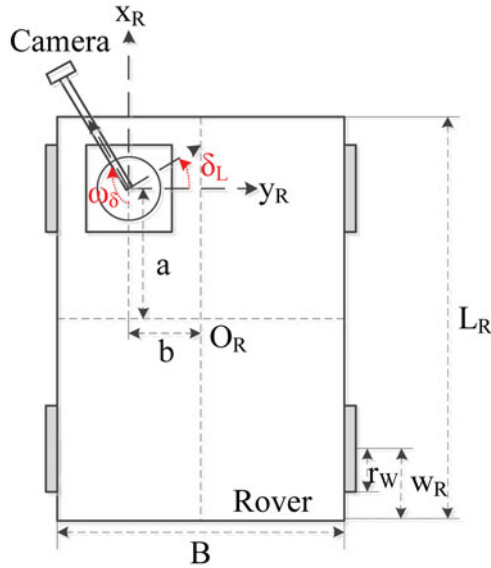


Figure 3. An installation model for the binocular camera (vertical view).

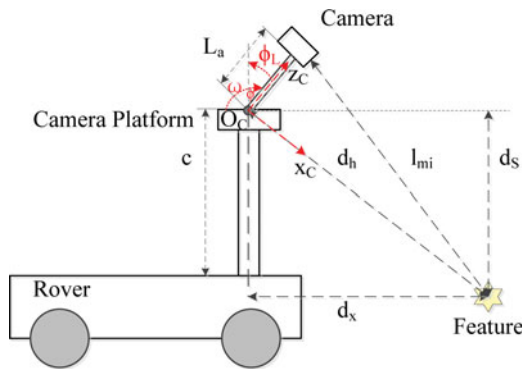


Figure 4. An installation model for the binocular camera (lateral view).

Valid feature points are selected according to measuring distances. In this paper, the  $i$ -th valid point is equivalent to the  $i$ -th feature point in the external environment. Aiming at the  $i$ -th feature point, the measurement model of the camera can be written as follows:

$$l_{mi} = \sqrt{(p_{L,x} - m_{i,x})^2 + (p_{L,y} - m_{i,y})^2 + (p_{L,z} - m_{i,z})^2} \tag{7}$$

where  $p_L$  is the fixed point of the binocular camera in the global coordinate system and  $m_i$  is the  $i$ -th feature point in the global coordinate system. The new feature point will be projected to the global coordinate system by the SLAM algorithm when a new feature point is observed. For each valid feature point, the position in the coordinate system of the

camera is given by:

$$m_{i_C} = \begin{bmatrix} m_{x_C,i} \\ m_{y_C,i} \\ m_{z_C,i} \end{bmatrix} = \begin{bmatrix} l_{m,i} \cos \psi_i \\ l_{m,i} \sin \psi_i \\ L_a \end{bmatrix} \quad i = 1, \dots, N \tag{8}$$

where  $\psi_i$  is the scanning azimuth angle for the  $i$ -th feature point in the coordinate system of the camera and  $N$  is the number of feature points at that moment. The rotation matrix  $C_C^H$  from the coordinate system of the camera to the coordinate system of the holder platform is:

$$C_C^H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_L & \sin \phi_L \\ 0 & -\sin \phi_L & \cos \phi_L \end{bmatrix} \tag{9}$$

The attitude transition from the coordinate system of the holder platform to the coordinate system of the rover is just one rotation around the z axis, thus the rotation matrix  $C_H^R$  is:

$$C_H^R = \begin{bmatrix} \cos \delta_L & \sin \delta_L & 0 \\ -\sin \delta_L & \cos \delta_L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

therefore, the position of the feature point in the global coordinate system can be described as:

$$m_{i_G} = p_R + C_R^G \left( C_H^R C_C^H m_{i_C} + \begin{bmatrix} a \\ b \\ c \end{bmatrix} \right) \quad i = 1, \dots, N \tag{11}$$

where  $p_R = [x_R, y_R, z_R]^T$  is the position of the rover in the global coordinate system and  $d = [a, b, c]$  is the position of the holder platform in the coordinate system of the rover.

### 3. STANDARD EKF-SLAM PROBLEM FOR THE ROVER.

3.1. *System state equation.* The state of the rover includes three position coordinates and three attitude angles. The state equation can be obtained by discretising the motion model of the rover, that is:

$$\mathbf{X}_G(k) = \mathbf{X}_G(k-1) + \Delta T \begin{bmatrix} C_R^G(k) & 0 \\ 0 & C_\omega^G(k) \end{bmatrix} \begin{bmatrix} v_R(k) + \tilde{v}_R \\ \omega_R(k) + \tilde{\omega}_R \end{bmatrix} \tag{12}$$

where  $\Delta T$  is a discrete interval time,  $\tilde{v}_R \sim N(0, Q_v)$  and  $\tilde{\omega}_R \sim N(0, Q_\omega)$  are control noise and  $k$  is the discrete time coefficient. The positions of the valid feature points in the local

maps are static, thus the recursive state equation for a new feature point is:

$$\mathbf{X}_{m_i}(k) = \mathbf{X}_{m_i}(k - 1) \tag{13}$$

It is assumed that the estimated values of the state variables and the covariance matrix are  $\hat{\mathbf{X}}(k)$  and  $\mathbf{P}(k)$  at time  $k$ , respectively:

$$\hat{\mathbf{X}}(k) = \begin{bmatrix} \hat{\mathbf{X}}_R(k) \\ \hat{\mathbf{X}}_m(k) \end{bmatrix} \tag{14}$$

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{RR}(k) & \mathbf{P}_{Rm}(k) \\ \mathbf{P}_{Rm}^T(k) & \mathbf{P}_{mm}(k) \end{bmatrix} \tag{15}$$

where  $\hat{\mathbf{X}}_R(k)$  is the estimated value of the position and attitude of the rover,  $\hat{\mathbf{X}}_m(k)$  is the estimated value of the feature point,  $\mathbf{P}_{RR}(k)$  is the covariance matrix of the rover,  $\mathbf{P}_{mm}(k)$  is the covariance matrix of the feature point and  $\mathbf{P}_{Rm}(k)$  is the cross covariance of the rover and the feature point.

The predicted value of the position and attitude of the rover at time  $k$  can be defined as:

$$\hat{\mathbf{X}}_R(k|k - 1) = \hat{\mathbf{X}}_R(k - 1) + \begin{bmatrix} \hat{\mathbf{C}}_R^G(k - 1) & 0 \\ 0 & \hat{\mathbf{S}}_\omega^G(k - 1) \end{bmatrix} \begin{bmatrix} T^2 \mathbf{a}_R(k) \\ T^2 \boldsymbol{\omega}_R(k) \end{bmatrix} \tag{16}$$

where  $\mathbf{a}_R(k)$  and  $\boldsymbol{\omega}_R(k)$  are the linear acceleration and angular velocity of the rover, assumed measured by an IMU and  $\hat{\mathbf{S}}_\omega^G(k - 1)$  is the transformation matrix of the angular velocity as in Equation (3). The predicted state variables and covariance matrix are then given as:

$$\hat{\mathbf{X}}(k|k - 1) = \begin{bmatrix} \hat{\mathbf{X}}_R(k|k - 1) \\ \hat{\mathbf{X}}_m(k - 1) \end{bmatrix} \tag{17}$$

$$\mathbf{P}(k|k - 1) = \nabla f_X(k) \mathbf{P}(k - 1) \nabla f_X^T(k) + \mathbf{Q}(k) \tag{18}$$

where  $\nabla f_X(k)$  is the Jacobian matrix to the state equation  $f(\cdot)$  and  $\mathbf{Q}(k)$  is the variance matrix of random state error. We assume  $\mathbf{Q}_{IMU}(k)$  is the covariance matrix of random state error generated by the IMU, thus:

$$\mathbf{P}(k|k - 1) = \nabla f_R(k) \mathbf{P}(k - 1) \nabla f_R^T(k) + \nabla f_{a,\omega}(k) \mathbf{Q}_{IMU} \nabla f_{a,\omega}^T(k) + \mathbf{Q}(k) \tag{19}$$

The predicted value of the whole covariance matrix can be simplified as:

$$\mathbf{P}(k|k - 1) = \begin{bmatrix} \nabla f_R(k) \mathbf{P}_{RR}(k - 1) \nabla f_R^T(k) + \mathbf{Q}_{RR}(k) & \nabla f_R(k) \mathbf{P}_{Rm}(k - 1) \\ \nabla f_R^T(k) \mathbf{P}_{mR}(k - 1) & \mathbf{P}_{mm}(k - 1) \end{bmatrix} \tag{20}$$

where  $\mathbf{Q}_{RR}(k)$  is the simplified form of the measurement noise variance matrix and state error variance, which can be described as:

$$\mathbf{Q}_{RR}(k) = \nabla f_{a,\omega}(k) \mathbf{Q}_{IMU} \nabla f_{a,\omega}^T(k) + \mathbf{Q}(k) \tag{21}$$



where the Jacobian matrix  $\nabla f_{a,\omega}(\mathbf{k})$  and  $\nabla f_R(\mathbf{k})$  are obtained as follows:

$$\nabla f_{a,\omega}(\mathbf{k}) = \begin{bmatrix} T^2 \hat{\mathbf{C}}_R^G(\mathbf{k}-1) & 0 \\ 0 & T \hat{\mathbf{S}}_\omega^G(\mathbf{k}-1) \end{bmatrix} \tag{22}$$

$$\nabla f_R(\mathbf{k}) = \frac{\partial \hat{\mathbf{X}}_R(\mathbf{k}|\mathbf{k}-1)}{\partial \hat{\mathbf{X}}_R(\mathbf{k}-1)} \tag{23}$$

$\mathbf{Q}_{IMU}$  is the covariance of the IMU:

$$\mathbf{Q}_{IMU} = \begin{bmatrix} \sigma_a^2 & \\ & \sigma_\omega^2 \end{bmatrix} \tag{24}$$

where  $\sigma_a$  is the standard deviation of the accelerometer’s random noise and  $\sigma_\omega$  is the standard deviation of the gyroscope’s random noise. At this moment, if new feature points are added to the global map, the EKF update process can be completed.

3.2. *Map augmentation.* It is assumed that the feature points have been obtained by a binocular camera, but the valid feature points cannot be brought into the state estimation process directly, since they need to be transformed into the global coordinate system. The observed variable  $\mathbf{z}_j(\mathbf{k})$  is associated to a new feature point  $m_j$  at time  $\mathbf{k}$ :

$$\mathbf{z}_j(\mathbf{k}) = \begin{bmatrix} l_{m_j}(\mathbf{k}) \\ \psi_j(\mathbf{k}) \end{bmatrix}^T = h_k(\mathbf{X}_G(\mathbf{k}), m_j) + \mathbf{v}_k \tag{25}$$

where  $\mathbf{v}_k \sim N(0, \mathbf{R}_{LL})$  is the gaussian white noise with zero mean and  $\mathbf{R}_{LL}$  is the variance. The position of the feature point  $m_j$  in the global coordinate system is:

$$m_{jG}(\mathbf{k}) = g_{kG}(\mathbf{z}_j(\mathbf{k}), \hat{\mathbf{X}}_G(\mathbf{k})) = \hat{\mathbf{P}}_R(\mathbf{k}) + \hat{\mathbf{C}}_R^G(\mathbf{k}) (\mathbf{C}_H^R \mathbf{C}_C^H m_{jC}(\mathbf{k}) + p_R) \tag{26}$$

where  $m_{jC}(\mathbf{k})$  is the position of the feature point in the coordinate system of the binocular camera, whose relation to the observed variable is:

$$m_{jC}(\mathbf{k}) = g_{kC}(\mathbf{z}_j(\mathbf{k})) = \begin{bmatrix} l_{m_j}(\mathbf{k}) \cos \psi_j(\mathbf{k}) \\ l_{m_j}(\mathbf{k}) \sin \psi_j(\mathbf{k}) \\ L_a \end{bmatrix} \tag{27}$$

Finally, the specific form of the observation equation becomes:

$$\hat{\mathbf{z}}_j = h_k(\hat{\mathbf{X}}_G, m_j) = \begin{bmatrix} \sqrt{(q_{xc})^2 (q_{yc})^2} \\ \arctan(q_{yc}/q_{xc}) \\ L_a \end{bmatrix} \tag{28}$$

$$q_C = \hat{\mathbf{C}}_H^C \hat{\mathbf{C}}_R^H \hat{\mathbf{C}}_G^R m_{jG} - \hat{\mathbf{C}}_H^C \hat{\mathbf{C}}_R^H (\hat{\mathbf{C}}_G^R \mathbf{p}_G + d_R^H)$$

where  $q_C$  represents the position of the feature point  $m_j$  in the coordinate system of the binocular camera.

So far, the augmentation process for the new feature points has been completed. Next, the augmentation for the variance matrix of the system state needs to be finished. The covariance matrix of the feature point itself is:

$$\begin{aligned}
 m_{j_G}(k) &= g_{k_G}(z_j(k), \hat{\mathbf{X}}_G(k)) \\
 \mathbf{G}_{\hat{\mathbf{X}}_G} &= \frac{\partial g_{k_G}(z_j(k), \hat{\mathbf{X}}_G(k))}{\partial \hat{\mathbf{X}}_G(k)} \\
 \mathbf{G}_{z_j} &= \frac{\partial g_{k_G}(z_j(k), \hat{\mathbf{X}}_G(k))}{\partial z_j}
 \end{aligned}
 \tag{29}$$

Then, the variance matrix of the new feature point and the covariance matrix of the planetary rover can be initialised as follows:

$$\begin{aligned}
 \mathbf{P}_{mm} &= \mathbf{G}_{\hat{\mathbf{X}}_G} \mathbf{P}_{RR} \mathbf{G}_{\hat{\mathbf{X}}_G}^T + \mathbf{G}_{z_j} \mathbf{R}_{LL} \mathbf{G}_{z_j}^T \\
 \mathbf{P}_{mx} &= \mathbf{G}_{\hat{\mathbf{X}}_G} \mathbf{P}_{Rx} = \mathbf{G}_{\hat{\mathbf{X}}_G} [\mathbf{P}_{RR} \quad \mathbf{P}_{\hat{\mathbf{X}}_G M}]
 \end{aligned}
 \tag{30}$$

where M in  $\mathbf{P}_{\hat{\mathbf{X}}_G M}$  is the initial collection of the map features, which is just a symbol for the initialisation map, the symbol L will be used to represent the global map when the features are added to the state equation. Finally, the augmentation matrices of the system state and variance become:

$$\begin{aligned}
 \mathbf{X}_R(k) &\rightarrow \begin{bmatrix} \mathbf{X}_R(k) \\ L_k = \begin{bmatrix} m_{1_G} \\ m_{2_G} \\ \vdots \\ m_{N_G} \end{bmatrix} \end{bmatrix} \\
 \mathbf{P}_{RR}(k) &\rightarrow \begin{bmatrix} \mathbf{P}_{RR}(k) & \mathbf{P}_{mx}^T \\ \mathbf{P}_{mx} & \mathbf{P}_{mm} \end{bmatrix}
 \end{aligned}
 \tag{31}$$

3.3. *State updates and revisit.* The predicted observed value  $\hat{z}(k|k - 1)$  is given in Section 3.2. When the observed value of the landmark feature is obtained by the sensor of the rover, the residual error  $\mathbf{r}(k)$  between real measured value  $\mathbf{z}(k)$  and predicted observed value  $\hat{z}(k|k - 1)$  can be expressed as:

$$\mathbf{r}(k) = \mathbf{z}(k) - \hat{z}(k|k - 1)
 \tag{32}$$

Therefore, the matrix  $\mathbf{S}(k)$  which is the covariance matrix of the residual error of the observed value is:

$$\mathbf{S}(k) = \nabla h_X(k) \mathbf{P}(k|k - 1) \nabla h_X^T(k) + \mathbf{Q}_v(k)
 \tag{33}$$

where  $\nabla h_X(k)$  is the Jacobian matrix of the observation model with respect to the state variables and  $\mathbf{Q}_v(k)$  is the covariance matrix of the observation model. It is assumed that only one feature point is observed during each observation process, then  $\nabla h_X(k)$  can be

expressed as:

$$\nabla h_X(k) = \left. \frac{\partial h(k)}{\partial \mathbf{X}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} = [\nabla h_R(k) \ 0 \ \dots \ 0 \ \nabla h_j(k) \ 0 \ \dots] \quad (34)$$

where  $\nabla h_R(k)$  is the Jacobian matrix of the observation model for the position and attitude variables of the rover and  $\nabla h_j(k)$  is the Jacobian matrix of the observation model for the feature point state.

After the observed variables are associated with the map state and the feature point state, the estimated state variables and the covariance matrix of the state variables can be updated by using the gain matrix  $\mathbf{W}(k)$ . The process is:

$$\hat{\mathbf{X}}(k) = \hat{\mathbf{X}}(k|k-1) + \mathbf{W}(k)\mathbf{r}(k) \quad (35)$$

$$\mathbf{P}(k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \quad (36)$$

$$\mathbf{W}(k) = \mathbf{P}(k|k-1) \nabla_X h^T(k)\mathbf{S}^{-1}(k) \quad (37)$$

The revisit process is similar to the update process of the EKF, which is the most important link.

It is assumed that the feature point  $m_{i_c}$  observed by the rover at time  $k$  is the feature point associated to the feature point  $m_{j_g}$  in the map, so the feature point  $m_{j_g}$  is added to the map at time  $(k - w)$ , where  $w$  denotes a certain time interval. Therefore, the rover is able to update the EKF state combining the information from  $m_{j_g}$  and  $m_{i_c}$ , which reduces the cumulative uncertainty of the state estimation from time  $(k - w)$  to time  $k$ .

4. SLAM WITH LOCAL SUBMAPS. The computational complexity is proportional to the square of the number of feature points contained in the state variables in a single filtering process. The local submap method can reduce the computational complexity of the SLAM by updating the covariance of whole system within a suitable interval (Wang et al., 2018). The local submaps are updated independently before joining the global map. Estimation of the local map is irrelevant to the estimation of the global map; therefore, the covariance of the local status and the global status is zero. The global estimation is not changed during the local update because it is irrelevant, which decreases the computational complexity during the prediction process. First, an independent local submap is established for the feature points observed by the camera in the early stage. Then, when the feature points contained in the submap reach a certain threshold value, namely, the submap reaches a certain scale, the local submap can be converted to the global state vector, which constitutes the global map. Next, the local submap is initialised for new feature points. Finally, a number of observations are consistently integrated into the global map after a time interval, which reduces the uncertainty of the state variables of the rover.

4.1. System state variables. The system state variables of the local submap method are:

$$\mathbf{X}_{IS}(k) = \begin{bmatrix} {}^G \mathbf{X}_S(k) \\ {}^G \mathbf{X}_i(k) \\ {}^S \mathbf{X}_R(k) \\ {}^S \mathbf{X}_i(k) \end{bmatrix} \quad (38)$$

where  ${}^G\mathbf{X}_S(k)$  is the position and attitude vector of local submap in the global map,  ${}^G\mathbf{X}_i(k)$  is the position of the feature point in the global map,  ${}^S\mathbf{X}_R(k)$  is the position and attitude vector of the rover in the local submap and  ${}^S\mathbf{X}_i(k)$  is the position of the feature point in the local submap.  ${}^G\mathbf{X}_m(k)$  and  ${}^S\mathbf{X}_m(k)$  can be used to represent the collection of the feature points in global map and the collection of the feature points in the local submap.

4.2. *System state estimation process.* The state covariance matrix includes the covariance of the global state variables and the local state variables and the cross covariance between them. However, the cross covariance between the global state variables and the local state variables is zero. Therefore, the state covariance matrix  $\mathbf{P}_{IS}(k)$  can be represented as:

$$\mathbf{P}_{IS}(k) = \begin{bmatrix} {}^G\mathbf{P}_{SS}(k) & {}^G\mathbf{P}_{mS}(k) & 0 & 0 \\ {}^G\mathbf{P}_{Sm}(k) & {}^G\mathbf{P}_{mm}(k) & 0 & 0 \\ 0 & 0 & {}^S\mathbf{P}_{RR}(k) & {}^S\mathbf{P}_{Rm}(k) \\ 0 & 0 & {}^S\mathbf{P}_{mR}(k) & {}^S\mathbf{P}_{mm}(k) \end{bmatrix} \tag{39}$$

where  ${}^G\mathbf{P}_{SS}(k)$  is the covariance matrix of the position estimation of the local submap in the global reference and  ${}^S\mathbf{P}_{RR}(k)$  is the covariance matrix of the position and attitude estimation of the rover in the local reference.

The predicted state value is then:

$$\mathbf{X}_{IS}(k|k-1) = \begin{bmatrix} {}^G\mathbf{X}(k-1) \\ f({}^S\mathbf{X}_R(k-1), \mathbf{u}(k)) \\ {}^S\mathbf{X}_m(k-1) \end{bmatrix} \tag{40}$$

The covariance matrix is updated by the rover model during the prediction process. The global estimation is not changed during the local update, because the local submaps are updated independently before joining the global map. The computational complexity is reduced because of the irrelevance.  $\mathbf{P}_{IS}(k|k-1)$  is given as:

$$\mathbf{P}_{IS}(k|k-1) = \begin{bmatrix} {}^G\mathbf{P}(k-1) & 0 & 0 \\ 0 & \nabla f_X(k) {}^S\mathbf{P}_{RR}(k-1) \nabla f_X^T(k) + \mathbf{Q}(k) & \nabla f_X(k) {}^S\mathbf{P}_{Rm}(k-1) \\ 0 & (\nabla f_X(k) {}^S\mathbf{P}_{Rm}(k-1))^T & {}^S\mathbf{P}_{mm}(k-1) \end{bmatrix} \tag{41}$$

The predicted state value  $\hat{z}(k|k-1)$  in the local submap method is:

$$\hat{z}(k|k-1) = h\left({}^S\hat{\mathbf{X}}_R(k-1), {}^S\hat{\mathbf{X}}_i(k-1)\right) \tag{42}$$

The observation residuals and the covariance are:

$$\mathbf{r}(k) = z(k) - \hat{z}(k|k-1) \tag{43}$$

$$\mathbf{S}(k) = \nabla h_X(k) {}^S\mathbf{P}(k|k-1) \nabla h_X^T(k) + \mathbf{Q}_v(k) \tag{44}$$

Finally, the update process can be written as:

$$\hat{\mathbf{X}}_{IS}(k) = \hat{\mathbf{X}}_{IS}(k|k-1) + \mathbf{W}(k)\mathbf{r}(k) \tag{45}$$

$$\mathbf{P}_{IS}(k) = \mathbf{P}_{IS}(k|k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \tag{46}$$

$$\mathbf{W}(k) = \mathbf{P}_{IS}(k|k-1) \nabla_X h^T(k) \mathbf{S}^{-1}(k) \tag{47}$$

The feature points contained in the local reference are far less than those contained in the global reference, therefore the computational complexity will be reduced significantly.  $\mathbf{P}_{lS}(\mathbf{k})$  and  $\mathbf{W}(\mathbf{k})$  are given as:

$$\mathbf{P}_{lS}(\mathbf{k}) = \begin{bmatrix} {}^G\mathbf{P}(\mathbf{k}|\mathbf{k}-1) & 0 \\ 0 & {}^S\mathbf{P}(\mathbf{k}|\mathbf{k}-1) - \mathbf{W}(\mathbf{k})\mathbf{S}(\mathbf{k})\mathbf{W}^T(\mathbf{k}) \end{bmatrix} \tag{48}$$

$$\mathbf{W}(\mathbf{k}) = {}^S\mathbf{P}(\mathbf{k}|\mathbf{k}-1) \nabla h_X^T(\mathbf{k})\mathbf{S}^{-1}(\mathbf{k}) \tag{49}$$

4.3. *The transformation from local submap to global map.* The submap needs to be transformed to the global map, when the submap reaches a certain scale. Then a new submap is established and initialised. It is assumed that  $T(\mathbf{k})$  is the transformation matrix from the local reference to the global reference.

$${}^G\hat{\mathbf{X}}_{lS}(\mathbf{k}) = \mathbf{T}_S^G(\mathbf{k})\hat{\mathbf{X}}_{lS}(\mathbf{k}) \tag{50}$$

$${}^G\mathbf{P}_{lS}(\mathbf{k}) = \nabla \mathbf{T}_S^G(\mathbf{k})\mathbf{P}_{lS}(\mathbf{k}) (\nabla \mathbf{T}_S^G(\mathbf{k}))^T \tag{51}$$

Global variables can be written as:

$${}^G\hat{\mathbf{X}}_{lS}(\mathbf{k}) = \begin{bmatrix} {}^G\hat{\mathbf{X}}_S(\mathbf{k}) \\ {}^G\hat{\mathbf{X}}_i(\mathbf{k}) \end{bmatrix} = \mathbf{T}_S^G(\mathbf{k}) \begin{bmatrix} {}^G\hat{\mathbf{X}}_S(\mathbf{k}) \\ {}^S\hat{\mathbf{X}}_i(\mathbf{k}) \end{bmatrix} \tag{52}$$

where  $\mathbf{T}_S^G(\mathbf{k})$  is the transformation matrix:

$$\mathbf{T}_S^G(\mathbf{k}) = \left[ \begin{array}{c|c} \mathbf{I}_6 & \mathbf{0} \\ \hline \mathbf{I}_3 & \mathbf{0} \end{array} \middle| \begin{array}{c} \mathbf{0} \\ \mathbf{C}_S^G(\mathbf{k}) \end{array} \right] \tag{53}$$

where  $\mathbf{C}_S^G(\mathbf{k})$  is the transformation matrix from the local coordinate system to the global coordinate system. The covariance matrix of the global variables can be expressed as:

$${}^G\mathbf{P}_{lS}(\mathbf{k}) = \nabla \mathbf{T}_S^G(\mathbf{k}) \begin{bmatrix} {}^G\mathbf{P}_{SS}(\mathbf{k}) & \\ & {}^S\mathbf{P}_{ii}(\mathbf{k}) \end{bmatrix} (\nabla \mathbf{T}_S^G(\mathbf{k}))^T \tag{54}$$

where  $\nabla \mathbf{T}_S^G(\mathbf{k})$  is the Jacobian matrix of the transformation matrix to the state variables, namely:

$$\nabla \mathbf{T}_S^G(\mathbf{k}) = \frac{\partial \mathbf{T}_S^G(\mathbf{k})}{\partial \left( {}^G\hat{\mathbf{X}}_S(\mathbf{k}), {}^S\hat{\mathbf{X}}_i(\mathbf{k}) \right)} \tag{55}$$

4.4. *Estimation constraints for local feature points.* The estimation of the feature points in the global coordinate system cannot be applied to the update process in the local coordinate system without the coordinate system connection. Moreover, local feature points will be computed multiple times when proceeding to the coordinate transformation process. In order to maintain the effectiveness of the algorithm, the independent estimation of the feature points should be combined and applied to the global estimation process.

The consistent estimation of the feature point status can be made through the known constraints between common states. The independent estimation of the feature point status is integrated into the global coordinate system by the constraints between the state variables, which generates a consistent global estimation result. The constraint process can be

simplified down to the weighted projection process of the crossing states, which can be described as follows:

$$\mathbf{C}(\hat{\mathbf{X}}(k)) = \mathbf{b} \quad (56)$$

An EKF algorithm can be used for approximate estimation, therefore the result of the constraint is:

$$\hat{\mathbf{X}}_c(k) = \hat{\mathbf{X}}(k) + \mathbf{W}_c(k) (\mathbf{b} - \mathbf{C}(k)\hat{\mathbf{X}}(k)) \quad (57)$$

$$\mathbf{P}_c(k) = \mathbf{P}(k) + \mathbf{W}_c(k)\mathbf{S}_c(k)\mathbf{W}_c^T(k) \quad (58)$$

where:

$$\mathbf{W}_c(k) = \mathbf{P}(k)\nabla\mathbf{C}(k)\mathbf{S}_c^{-1}(k) \quad (59)$$

$$\mathbf{S}_c(k) = \nabla\mathbf{C}(k)\mathbf{P}(k)\nabla\mathbf{C}^T(k) \quad (60)$$

The constraints can be described as:

$$\mathbf{C}(k) = \begin{bmatrix} {}^G\mathbf{X}_S(k) \\ {}^G\mathbf{X}_i(k) \\ {}^S\mathbf{X}_R(k) \\ {}^S\mathbf{X}_i(k) \end{bmatrix} = \mathbf{0} \quad (61)$$

where:

$$\mathbf{C}(k) = [-\mathbf{I} \quad \mathbf{I} \quad \mathbf{0} \quad -\mathbf{I}] \quad (62)$$

The process of transforming local feature points to the global map can be filtered by Equations (57)–(60), where  $\mathbf{b} = \mathbf{0}$ .

**5. BODY-FIXED SLAM WITH LOCAL SUBMAPS.** The errors in the linearization process of the observation equation and rover's motion model are the main cause of the inconsistency. Therefore, reducing the linearization error is the important research content, which can reduce the uncertainty of the estimation process and increase the consistency of the algorithm.

The rover body-fixed SLAM algorithm with local submaps is presented in this section. The traditional EKF-SLAM algorithm is established in the absolute coordinate system, which increases the linearization error, due to the accumulative uncertainty of the rover's pose and the accumulative uncertainty of the feature points' positions which are introduced into the observation equation. The feature points will be updated in the coordinate system of the planetary rover in the body-fixed SLAM algorithm, namely, only the uncertainty of the binocular camera and IMU observed noise will be introduced into the linearization process of the observation equation, which will reduce the linearization errors. In addition, a better result can be obtained by combining the local submaps and transformation of the coordinates.

5.1. *The state prediction process.* The coordinate system of the rover is considered as the reference coordinate in this section. The state variables of the system can be described as follows:

$${}^R\mathbf{X} = \begin{bmatrix} {}^R\mathbf{X}_B \\ {}^R\mathbf{X}_1 \\ \vdots \\ {}^R\mathbf{X}_{n_f} \end{bmatrix} \quad {}^R\mathbf{P} = \begin{bmatrix} {}^R\mathbf{P}_{BB} & \cdots & {}^R\mathbf{P}_{Bn_f} \\ \vdots & \ddots & \vdots \\ {}^R\mathbf{P}_{n_f B} & \cdots & {}^R\mathbf{P}_{n_f n_f} \end{bmatrix} \quad (63)$$

where  ${}^R\mathbf{X}_B$  is the vector of the absolute coordinate in the coordinate system of the rover,  ${}^R\mathbf{P}_{BB}$  is the covariance of  ${}^R\mathbf{X}_B$ ,  $n_f$  denotes the number of the feature points and  ${}^R\mathbf{X}_i, i = 1, \dots, n_f$  is the position of the feature points in the coordinate system of the rover.

The position of the feature point in the body coordinate at time  $k$  can be written as:

$${}^R\mathbf{X}_i(k) = {}^R\mathbf{X}_{k-1}^k(k) \oplus {}^R\mathbf{X}_i(k-1) \quad (64)$$

where  $\oplus$  denotes the vector sum including coordinate transformation and position shift. The pose observed by the IMU is considered as an independent feature point, which is added to the state variables. The predicted values of the state variables and covariance matrix can be described as follows:

$$\begin{aligned} {}^R\hat{\mathbf{X}}(k|k-1) &= \begin{bmatrix} {}^R\hat{\mathbf{X}}(k-1) \\ {}^R\hat{\mathbf{X}}_{k-1}^k(k|k-1) \end{bmatrix} \\ {}^R\mathbf{P}(k|k-1) &= \begin{bmatrix} {}^R\hat{\mathbf{P}}(k-1) & \\ & \mathbf{Q}_{IMU} \end{bmatrix} \end{aligned} \quad (65)$$

where  $\mathbf{Q}_{IMU}$  is the covariance matrix of the observation noise.

5.2. *The state update process.* The positions of the feature points need to be transferred into the body coordinates from global coordinates. The estimated value of the observation  $\hat{z}(k|k-1)$  is:

$$\hat{z}(k|k-1) = h\left({}^R\hat{\mathbf{X}}_i, {}^R\hat{\mathbf{X}}_{k-1}^k\right) \quad (66)$$

The residual error  $\mathbf{r}(k)$  between the real measured value  $\mathbf{z}(k)$  and predicted value  $\hat{z}(k|k-1)$  is:

$$\mathbf{r}(k) = \mathbf{z}(k) - \hat{z}(k|k-1) \quad (67)$$

The covariance matrix of the residual error of observation variable can be described as:

$$\mathbf{S}(k) = \nabla h_{R_x}(k)\mathbf{P}(k|k-1)\nabla h_{R_x}^T(k) + \mathbf{Q}_v(k) \quad (68)$$

The whole state update process is similar to those in previous sections. The covariance matrix after augmentation is shown as follows, which should be noted:

$${}^R\mathbf{P}(k) = \begin{bmatrix} {}^R\mathbf{P}(k|k-1) & \\ & \mathbf{Q}_v \end{bmatrix} \quad (69)$$

5.3. *The state merging process.* The position of the feature points and global coordinate in the body coordinate should be updated by pose changes.

$${}^R\hat{\mathbf{X}}_B(k) = {}^R\hat{\mathbf{X}}_B(k-1) \oplus {}^R\hat{\mathbf{X}}_{k-1}^k(k) \tag{70}$$

$${}^R\hat{\mathbf{X}}_i(k) = {}^R\hat{\mathbf{X}}_i(k-1) \oplus {}^R\hat{\mathbf{X}}_{k-1}^k(k) \tag{71}$$

Meanwhile the update process of the state covariance can be obtained as:

$${}^R\mathbf{P}_{BB}(k) = \nabla^R\hat{\mathbf{X}}_B \begin{bmatrix} {}^R\mathbf{P}_{BB}(k) & \\ & {}^R\mathbf{P}_{\hat{\mathbf{X}}_{k-1}^k}(k) \end{bmatrix} \left( \nabla^R\hat{\mathbf{X}}_B \right)^T \tag{72}$$

$${}^R\mathbf{P}_{ii}(k) = \nabla^R\hat{\mathbf{X}}_i \begin{bmatrix} {}^R\mathbf{P}_{ii}(k) & \\ & {}^R\mathbf{P}_{\hat{\mathbf{X}}_{k-1}^k}(k) \end{bmatrix} \left( \nabla^R\hat{\mathbf{X}}_i \right)^T \tag{73}$$

where  $\nabla^R\hat{\mathbf{X}}_B$  and  $\nabla^R\hat{\mathbf{X}}_i$  are covariance matrices of the estimated variables  ${}^R\hat{\mathbf{X}}_B$  and  ${}^R\hat{\mathbf{X}}_i$  during the update process.  $\nabla^R\hat{\mathbf{X}}_B$  can be described as follows:

$$\nabla^R\hat{\mathbf{X}}_B = \begin{bmatrix} \frac{\partial^R\hat{\mathbf{X}}_B(k)}{\partial^R\hat{\mathbf{X}}_B(k-1)} & \frac{\partial^R\hat{\mathbf{X}}_B(k)}{\partial^R\hat{\mathbf{X}}_{k-1}^k(k)} \end{bmatrix} \tag{74}$$

the merging process can be done through Equations (71)–(74).

5.4. *The constraints for body-fixed SLAM with submaps.* The state variables and covariance matrix are defined as follows, corresponding to the local submaps method:

$${}^R\mathbf{X}_l = \begin{bmatrix} {}^R\mathbf{X}_G \\ {}^R\mathbf{X}_s \end{bmatrix} \quad {}^R\mathbf{P}_l = \begin{bmatrix} {}^R\mathbf{P}_G & \\ & {}^R\mathbf{P}_s \end{bmatrix} \tag{75}$$

the submaps will be integrated into the body-fixed global map when the submaps reach a certain scale, and the process is shown in Section 3.

Only the changes in state variables and relative positions of the feature points are updated in the body-fixed SLAM algorithm. Equation (75) can be written as follows:

$${}^R\mathbf{X}_l = \begin{bmatrix} {}^R\mathbf{X}_B^G \\ {}^R\mathbf{X}_i^G \\ {}^R\mathbf{X}_l^S \\ {}^R\mathbf{X}_i^S \end{bmatrix} \quad {}^R\mathbf{P}_l = \begin{bmatrix} {}^R\mathbf{P}_G & \\ & {}^R\mathbf{P}_s \end{bmatrix} \tag{76}$$

where  ${}^R\mathbf{X}_l^S$  is the accumulation values of the state changes in the submaps.  ${}^R\mathbf{X}_i^G$  and  ${}^R\mathbf{X}_i^S$  represent the same feature point in the body-fixed coordinate and local submap, then we have the following constraint:

$$\mathbf{c}^R\mathbf{X}_i = \mathbf{b} \tag{77}$$

where  $\mathbf{c} = [0 \quad \mathbf{I} \quad \mathbf{I} \quad -\mathbf{I}]$   $\mathbf{b} = 0$ , because the same feature point should be the same by coordinate transformation in different coordinate systems. The filtering process was illustrated in Section 3.



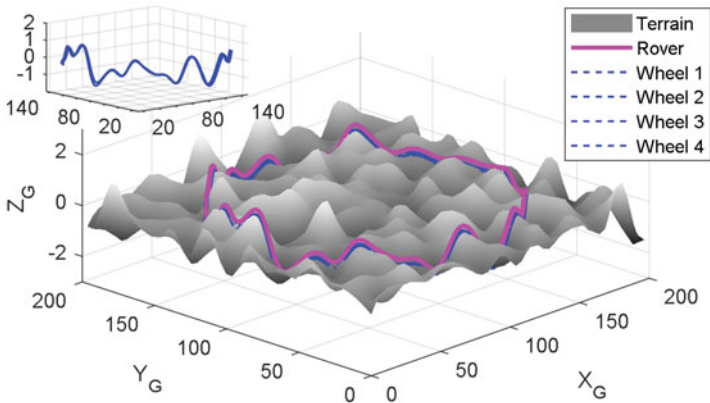


Figure 5. Three-dimensional trajectory of the planetary rover with the terrain constraint.

Table 1. The setting of simulation parameters.

Name	Value
Sampling period	0.2 s
The sum of feature points in the map	200
The radius of wheel	0.2 m
The initial position of the rover	[100, 20, -0.3174]
The SD of RN of accelerometer	$2.0 \times 10^{-3} \text{ m/s}$
The SD of RN of gyroscope	$0.02^\circ/\text{s}$
The SD of RN of measuring length	1.0 m
The SD of RN of measuring azimuth	$4^\circ$
The SD of RN of control stick	$4^\circ$
Sensor frequency of IMU	20 Hz
Sensor frequency of camera	0.5 Hz
Simulation time	400 s

Notice: SD denotes Standard Deviation, RN denotes Random Noise.

6. SIMULATION RESULTS. Simulation results are analysed in this section. It is assumed that the trajectory of the planetary rover is a circle, so that the modified EKF-SLAM can be analysed more conveniently. The Three-Dimensional (3D) trajectory of the planetary rover with the terrain constraint is shown in Figure 5, where the solid pink line is the trajectory of the centre of mass of the rover and the blue dotted lines are the trajectories of the contact point of the wheels. Meanwhile, the trajectories of the contact point of the wheels are enlarged in the upper-left figure in blue dotted lines.

200 landmarks represented as asterisks are distributed randomly in the map. The simulation parameters used in the following simulation process are shown in Table 1.

The true path and the estimated trajectory by standard EKF-SLAM are shown in Figure 6, where the bold pink line represents the planetary rover’s planned trajectory and the fine blue line represents the estimated result by EKF-SLAM. The green points denote feature points in the map, while the red points denote the observed feature points and the black ellipses around the red points denote the uncertainty ellipses.

The position error curve and the attitude error curve of the rover based on standard EKF-SLAM are shown in Figures 7 and 8, respectively, which are drawn in blue lines.

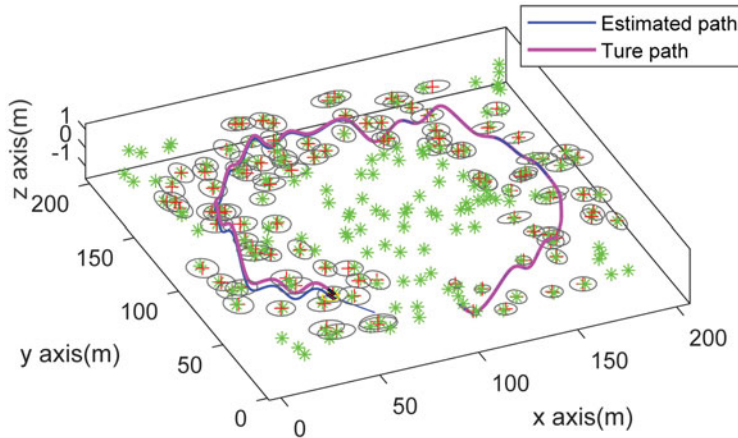


Figure 6. Simulation results of the true path and the estimated trajectory by standard EKF-SLAM algorithm.

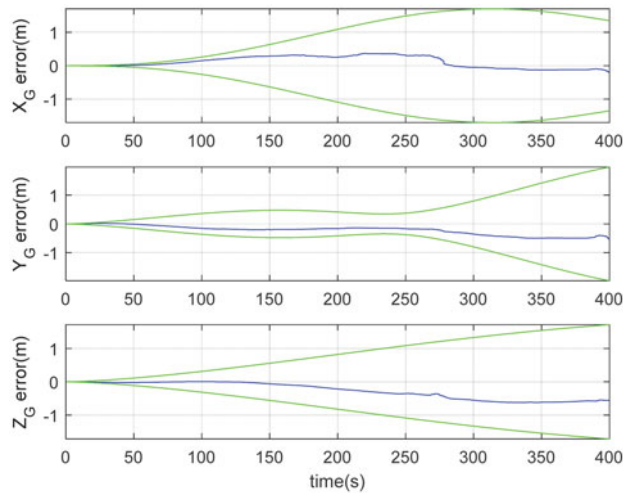


Figure 7. The position error curve of the rover (Standard EKF-SLAM).

Meanwhile, the green lines in Figures 7 and 8 are the covariance curves of the variables, namely, position covariance curve and attitude covariance curve.

The position error is less than 1 m, and the attitude error is less than  $0.5^\circ$ , which can be analysed through the following simulation results. The trace of the covariance of feature points is shown in Figure 9, from which a new feature point observed by the rover can be seen; the trace of feature points increases immediately.

In the following part, the body-fixed SLAM algorithm with local submaps is analysed. The actual motion trajectory of the rover and the sensor parameters are the same as above in Table 1. The true path and the estimated trajectory based on body-fixed SLAM with local submaps are shown in Figure 10, where the bold pink line denotes the actual trajectory and the fine blue line denotes the estimated trajectory. The green points denote feature points

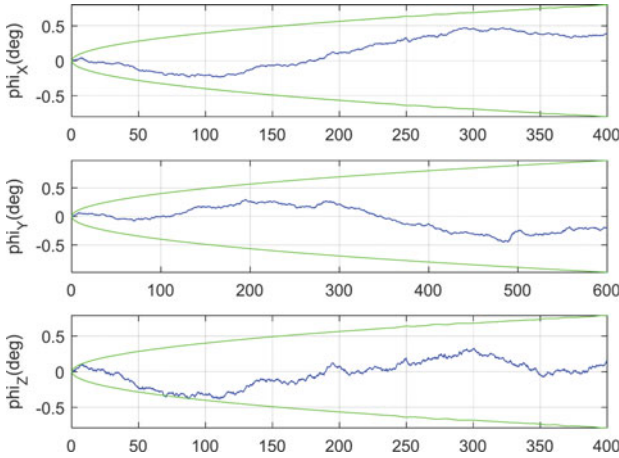


Figure 8. The attitude error curve of the rover (Standard EKF-SLAM).

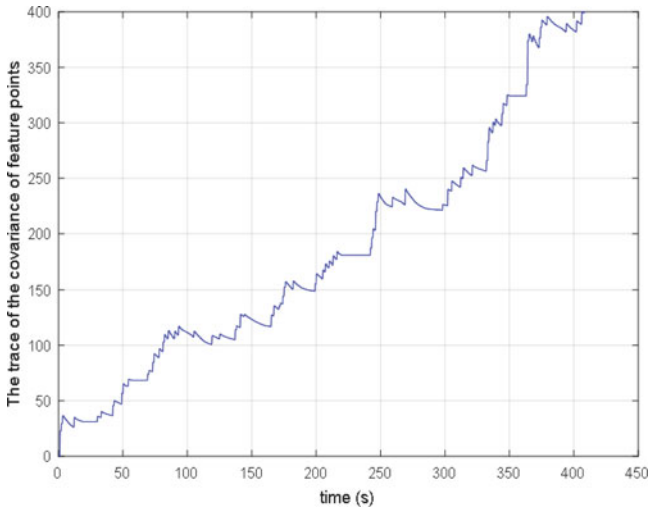


Figure 9. The trace of the covariance of feature points (Standard EKF-SLAM).

in the map, while the red points denote the observed feature points and the black ellipses around the red points denote the uncertainty ellipses.

The position error curves and the attitude error curves of the rover based on body-fixed SLAM are shown in Figures 11 and 12, respectively, which are drawn in blue lines. Meanwhile, the green lines in Figures 11 and 12 are the covariance curves of the variables, namely, position covariance curve and attitude covariance curve.

The estimated error of position is kept within 1 m while the estimated error of the attitude is kept within  $0.5^\circ$ , which are shown in Figures 11 and 12. The accuracy of body-fixed SLAM is close to standard EKF-SLAM. Moreover, the uncertainty ellipses are smaller in Figure 8 than those in Figure 6. Therefore, the uncertainty of the rover's motion in the global map can be reduced because of the constraint filtering and local submaps.

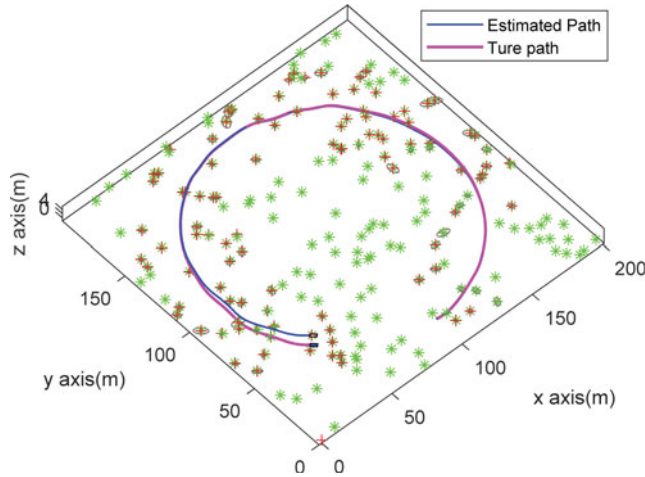


Figure 10. Simulation results of the true path and the estimated trajectory by body-fixed SLAM algorithm with local submaps.

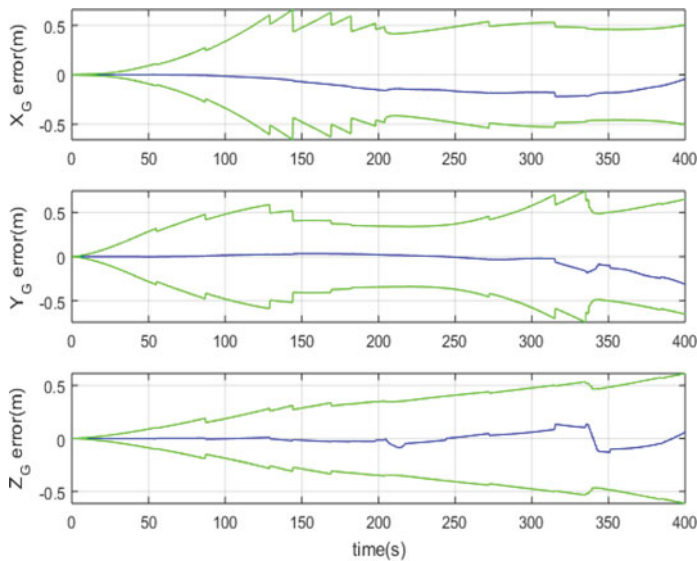


Figure 11. The position error curve of the rover (Body-fixed SLAM).

The Standard Deviation (SD) curve of the local position and azimuth deviation of the rover is shown in Figure 13. The SD will decrease after each period of time using the body-fixed SLAM algorithm compared with the standard EKF-SLAM algorithm in Figure 9. The cumulative values of the feature points contained in the state variables of the standard EKF-SLAM and body-fixed SLAM with local submaps are shown in Figure 14.

The computational complexity is proportional to the square of the number of feature points contained in the state variables. Therefore, the computational complexity is reduced significantly in body-fixed SLAM. This is because only the local submaps which contain

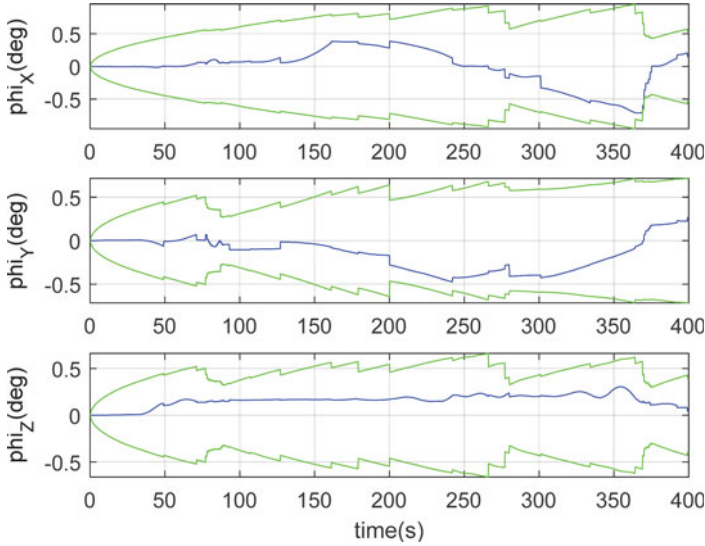


Figure 12. The attitude error curve of the rover (Body-fixed SLAM).

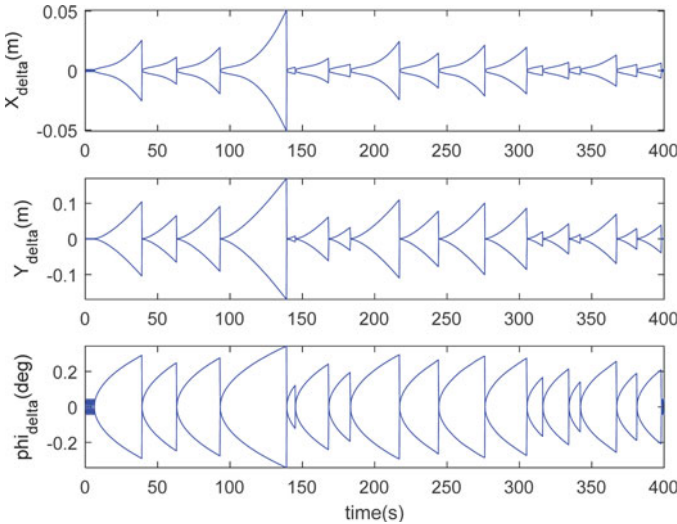


Figure 13. The Standard Deviation (SD) curve of the local position and azimuth deviation of the rover (Body-fixed SLAM).

minor feature points need to be updated in a single filtering process. In addition, the computational efficiency will be improved correspondingly. Some comparison results of the two algorithms are listed in Table 2, from which we can see the difference of the two algorithms clearly. The proposed method has a better performance.

7. CONCLUSIONS. In this paper, a body-fixed SLAM method with local submaps for planetary surface exploration is proposed. The changes of the rover’s position are taken

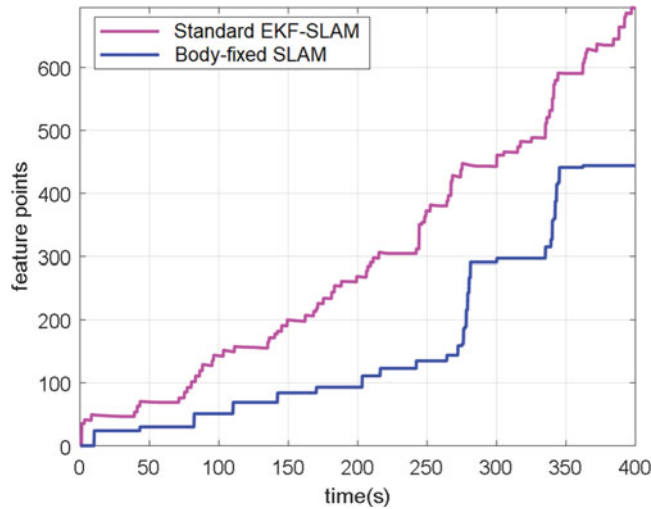


Figure 14. The cumulative values of the feature points contained in the state variables of standard EKF-SLAM and body-fixed SLAM with local submaps.

Table 2. Some comparison results of the two algorithms.

	Standard EKF-SLAM	Body-fixed SLAM with Submaps
The total number of the feature points contained in the state variables	700	440
Computational complexity	$\propto (700)^3$	$\propto (440)^3$
The triaxial position error of rover	$\leq 1$ m	$\leq 0.5$ m
The triaxial angle error of rover	$\leq 0.5^\circ$	$\leq 0.5^\circ$
Time cost	69 s	41 s

as the state variables before the update of EKF, and then the process of update for EKF is started. Finally, the position of feature points in the body-fixed coordinate is updated. The error of the position and attitude can be controlled within acceptable ranges (1 m,  $0.5^\circ$ ) for a planetary rover by the body-fixed SLAM with submaps, which is shown in the simulation results. However, lower computational complexity is obtained through the joint algorithm presented in this paper. The algorithm can be used for planetary rovers in exploration missions. In future research, the algorithm represented in this paper is also expected to be used in multi-spacecraft formation exploration missions (Zhuo et al., 2018a; 2018b; 2018c).

#### ACKNOWLEDGEMENTS

This paper has been completed with the guidance and help of Professor Franco Bernelli-Zazzera and Professor Mauro Massari in Aerospace Engineering Department of Politecnico di Milano, and the authors would like to express their gratitude. This work is supported by the National

Science Foundation of China (No.61374213, No.61573247) and Project Supported by Key Laboratory Opening Funding of Landing and Returning Control Technique of Deep Space Exploration (HIT.KLOF.2018072).

## REFERENCES

- Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H.F. and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation*. **17**(3), 229–241.
- Islam, M. R., Chowdhury, F.H., Rezwan, S., Ishaque, M.J., Akanda, J.U., Tuhel, A.S. and Riddhe, B.B. (2017). Novel design and performance analysis of a Mars exploration robot: Mars rover mongol pothik. *In Research in Computational Intelligence and Communication Networks (ICRCICN)*. 132–136. Kolkata, India.
- Joly, C. and Rives, P. (2009). Building consistent local submaps with omnidirectional SLAM. *Computer Vision Workshops, 2009 IEEE 12th International Conference on IEEE*. 2180–2187. Kyoto, Japan.
- Leonard, J.J. and Durrant-Whyte, H.F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on robotics and Automation*. **7**(3), 376–382.
- Martinez-Cantin, R. and Castellanos, J.A. (2006). Bounding uncertainty in EKF-SLAM: The robocentric local approach. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on IEEE*. 430–435. Orlando, FL, USA.
- Meng, Y. and Hutao C. (2014). A new approach based on crater detection and matching for visual navigation in planetary landing. *Advances in Space Research*. **53**(12), 1810–1821.
- Martinez-Cantin, R. and Castellanos, J.A. (2005). Unscented SLAM for large-scale outdoor environments. *Intelligent Robots and Systems. 2005 IEEE/RSJ International Conference on. IEEE*. Edmonton, Alta, Canada.
- Nüchter, A., Lingemann, K., Hertzberg, J. and Surmann, H. (2007). 6D SLAM—3D mapping outdoor environments. *Journal of Field Robotics*. **24**(8–9), 699–722.
- Paz, L.M. and Neira, J. (2006). Optimal local map size for EKF-based SLAM. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on IEEE*. 5019–5025. Beijing, China.
- Tan, W., Liu, H., Dong, Z., Zhang, G. and Bao, H. (2013). Robust monocular SLAM in dynamic environments. *In Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium*. 209–218. Adelaide, SA, Australia.
- Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J. and Nebot, E. (2004). Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, **4**(3), 380–407.
- Wang, J., Song, J., Zhao, L. and Huang, S. (2018). A Submap Joining Based RGB-D SLAM Algorithm Using Planes as Features. *In Field and Service Robotics*. 367–382. ETH Zurich, Switzerland.
- Yen, J., Jain, A. and Balam, J. (1999). ROAMS: Rover Analysis Modeling and Simulation Software. *In Artificial Intelligence, Robotics and Automation in Space*, 440, 249. Noordwijk, Netherland.
- Zhuo, Z., Yang, S., Zexu, Z. and Weisheng, Y. (2018a). New Results on Sliding-Mode Control for Takagi-Sugeno Fuzzy Multiagent Systems. *IEEE Transactions on Cybernetics*. (99), 1–13.
- Zhuo, Z., Zexu, Z. and Hui Z. (2018b). Distributed Attitude Control for Multi-Spacecraft via Takagi-Sugeno Fuzzy Approach. *IEEE Transactions on Aerospace and Electronic Systems*. **54**(2), 642–654.
- Zhuo, Z., Yang, S., Zexu, Z., Hui, Z. and Sheng, B. (2018c). Modified Order-Reduction Method for Distributed Control of Multi-Spacecraft Networks With Time-Varying Delays. *IEEE Transactions on Control of Network Systems*, **5**(1), 79–92.