A navigation simulation system of lunar rover

Ji Liu\textsuperscript{1,2}, Yuechao Wang\textsuperscript{2}, Chuan Zhou\textsuperscript{2}, Yanfeng Geng\textsuperscript{1,2}

\textsuperscript{1}Graduate School of the Chinese Academy of Sciences, Beijing 100039, China
\textsuperscript{2}Robotics Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences Shenyang 110016, China

Abstract—This paper presents a navigation simulation system on lunar rover. With virtual reality technology, this system serves to verify the kinematics model and the validity of the planning-path for navigating lunar rover. Because the lunar terrain can not be treated as the 2D surface, the kinematics modeling of lunar rover unlike ordinary mobile robots needs to contain the information about terrain. This model assumes that the point of wheel-terrain point changes continuously. However, the terrain geometry and discretization in simulation do not meet this assumption so that error will destroy the simulation effect. An optimization strategy is proposed to eliminate this error and keeps tight contact between the wheels and the terrain. Finally, experiments validate the optimization method and the whole simulation system.

Index Terms—kinematics model, lunar rover, rover-terrain interaction, navigation simulation.

I. INTRODUCTION

Simulation provides an effective method to secure space robots such as lunar rover. Simulation can verify the physical models, the planning-path for robots, and so on. Thus, a large number of researches focus on simulation \cite{1,2,12,13,14}. In \cite{1}, M. Lacagnina et al. model and simulate a multibody mobile robot (M6) for volcanic environment explorations. In \cite{12}, G. Sohl and A Jain describe wheel-terrain contact modeling in the ROAMS physics-based simulator for planetary surface exploration rover vehicles. In \cite{13}, A. Jain et al. describe development of the ROAMS physics-based simulator for planetary surface exploration rover vehicles. In \cite{14}, Biesiadecki, J et al. introduce a real-time spacecraft dynamics simulator.

Our system mainly simulates the real-time motion state of lunar rover on lunar surface according to the planning-path and concludes whether this path predefined is feasible for it.

The structure of lunar rover shown in Fig. 1 is similar with that of ROVER7 and ROVER9 in \cite{3,8}.

Generally speaking, kinematics or dynamics model is the basis in robot simulation. Our simulation mainly depends on the kinematics model. The kinematics model describes the relationship between the rover pose difference and the 3D terrain difference.

The kinematics model of lunar rover differs from ordinary mobile robots \cite{5,6,7,9,10,11}. The kinematics modeling approach for ordinary mobile robots is either geometric \cite{5,6} or transformation \cite{7}. In their model, the terrain is considered as 2D space, so the model does not include any terrain information. Because the terrain on lunar terrain can not be described approximately as 2D surface, unlike them kinematics modeling for lunar rover needs to contain 3D information. Some assumption in model leads that this model is not accurate utterly in simulation. Furthermore, even though the model were absolutely accurate, the error can not be avoided because discretization would bring simulation with error. The error may influence the simulation effect. For example, a common phenomenon is that the wheels of rover gradually stick in or fly above the terrain in the process of simulation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{Lunar rover}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Simulation procedure}
\end{figure}
To solve this problem, we adopt an optimization method to eliminate error. This method defines an objective function which requires the rover wheels are as close to the terrain surface as possible.

The simulation procedure is presented as Fig. 2. At first, rover structure data, planning-path, initial pose parameters and 3D scene data are inputted into kinematics model. According to the model, the next step pose parameters of rover are calculated. According to the method for ordinary mobile robots, the pose parameters calculated through model will directly be used to 3D display. Just as what mentioned above, because of the model imprecision and discretization, the error will destroy the 3D display effect. Thus, in our simulation system, the optimization section is added to eliminate error. The parameters adjusted by optimization section become the final result for 3D display. In the next circle, the rover contemporary pose information will be sent to kinematics model for calculating next pose.

In addition, we assume the terrain surface is rigid in both kinematics model and 3D display.

This paper is arranged as follows: in section 2, we simply introduce the kinematics model; in section 3, some main reasons causing error will be analyzed and the optimization approach to eliminate error will be presented; in section 4, experiments about both synthetic and real 3D reconstructed scenes validate the effectiveness of this system and the optimization approach; the last section is the conclusion.

II. KINEMATICS MODEL

For space constraint, this paper does not plan to interpret detailedly the kinematics model. Our method is similar to the one in [3,4]. Hence, we only explain how the model contains the terrain information.

Since the rover moves on the 3D terrain, its pose is determined by both rover mechanics and the terrain geometry structure. The contact points between the rover wheels and the terrain are the key to link the rover and the terrain.

In our analysis, a single, rigid, continuous contact point between each wheel and the terrain is assumed. In order to capture the wheel motion, a contact coordinate frame $C_i (i = 1,..,6)$ is defined at each wheel contact points as illustrated in Fig.3, where its $x$-axis is tangent to the terrain at the point of contact and its $z$-axis is normal to the terrain. The contact angle $\delta_i$ is the angle between the $z$-axes of the $i$th wheel axle and contact coordinate frame, as shown in figure 3. It is the contact angle that links the rover pose with the terrain information. It is a key distinction between an ordinary mobile robot moving on flat surface and the rover traversing uneven terrain.

Through a series of deduction, we finally get the relationship between the parameters about rover and planning-path as well as terrain information as (1):

$$
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z} \\
\dot{\phi} \\
\dot{\theta} \\
\dot{\beta} \\
\dot{\rho}_1 \\
\dot{\rho}_2 \\
\dot{\omega} \\
\dot{\varepsilon}
\end{bmatrix} = T
\begin{bmatrix}
\delta \\
\dot{x} \\
\dot{y} \\
\dot{\psi}
\end{bmatrix}
$$

where $[x, y, z]$ are the position parameters, $[\phi, \theta, \psi]$ are the heading angle, pitch angle and roll angle, $\omega$ is the vector consisting of six wheels’ angular rotation, $\varepsilon$ is the vector of six wheels’ roll slip, $[\beta, \rho_1, \rho_2]$ are the parameters of rover joints, $\delta = [\delta_1, \delta_2, .., \delta_6]$ and $T$ is the transformation matrix. $\delta$ is decided by the rover pose and the terrain. $[x, y, \psi]$ can be established by the planning-path predefined.

Based on equation (1), we can calculate the rover pose after discretization as equation (2).

$$
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z \\
\Delta \phi \\
\Delta \theta \\
\Delta \beta \\
\Delta \rho_1 \\
\Delta \rho_2 \\
\Delta \omega \\
\Delta \varepsilon
\end{bmatrix} = T
\begin{bmatrix}
\delta \\
\Delta x \\
\Delta y \\
\Delta \psi
\end{bmatrix}
$$

where $n$ is the discrete time.

Fig. 3. Coordinate frame for terrain contact at the $i$th wheel

III. OPTIMIZATION APPROACH

A. Analysis of error cause

Some factors render the kinematics model (1) imprecise in simulation.

Firstly, the terrain geometry structure may not meet the assumption that wheel-terrain point change continuously. In Fig. 4, when the terrain is uneven severely, the wheel-terrain point tends to change largely from one step to next step.

Secondly, even though the terrain were almost even, the discretion of discretion can still lead to simulation error. In the simulation, the terrain and the time are unexceptionally discrete. If the terrain is even, the contact angle $\delta_i$ difference between two steps can approximately satisfy the condition that $\Delta \delta_i$ is a micro value. As Fig.5, when the burst appears in the terrain, the condition above can not be met so that the error in simulation will appear and gradually accumulate with time. In the practical simulation, the burst in Fig. 5 is hard to avoid since the terrain reconstruction can easily generate the noise.

Fig. 4 Discontinuous change of the wheel-terrain contact point on uneven terrain
B. Optimization

For those reasons above, purely based on the kinematics model, the simulation error will influence the simulation precision. In the experiences, we find that when error emerges, the most apparent phenomena are that the wheels of rover tend to stick into the 3D terrain and fly above the terrain. We call it as the error of interaction between the wheels and the terrain.

Thus, we adopt the optimization approach to eliminate it. Since the error emerges in the relationship between the wheel and the terrain, we design an energy function to minimize the error of interaction. Of course, the kinematics model should be satisfied to the most extent.

The pose and position of rover in the 3D terrain can be established by nine parameters: \([x, y, z, \phi, \theta, \psi, \beta, \rho_1, \rho_2]\), in which, \([x, y, \psi]\) are predefined according to planning-path. Actually, only six parameters can be adjusted. In order that the rover can contact the terrain tightly, the objective function is described as equation (3):

\[
J_i = \overline{D} \Sigma_i \overline{D}^T
\]  

where \(\overline{D} = [D_1, D_2, \ldots, D_n]\), \(D_i = D_i(\overline{p})\) is the distance between the ith wheel contact point and the terrain and \(D_i\) can be established by the rover pose parameters and the 3D terrain information \(\overline{p} = [z, \phi, \theta, \psi, \beta, \rho_1, \rho_2]\), \(\Sigma_i\) is the weight diagonal matrix which represents the strength of penalty for each distance. In addition, the kinematics model can not be abandoned, because it describes accurately the relationship among all joints of rover pose, and the general relationship between the 3D terrain and rover pose. The rover pose estimation should satisfy this model to some extent. Therefore, another part as equation (4) is necessary.

\[
J_2 = (\overline{p} - \overline{p}_m) \Sigma_2 (\overline{p} - \overline{p}_m)^T
\]  

where \(\overline{p}_m\) is the vector of parameters through calculating kinematics model; \(\Sigma_2\) is the weight diagonal matrix which represents the degree of constraint for each parameter. This function ensures that the parameter is consistent with this model result. Moreover, only minimizing \(J_1\) may lead to non-unique solution, so a conclusion criterion to decide the best solution is necessary. Kinematics model provides a measurement for it. The best solution should be close to the solution deduced by the model.

Finally, the objective function is defined as (5):

\[
J = J_1 + \lambda J_2
\]  

where \(\lambda\) is a constant which represents the constraint degree from kinematics model.

In addition, \(J_1\) can be improved in the practical simulation. Just as what mentioned above, the main error exhibition in the simulation can be divided into two main types. One is that the wheels run into the terrain; the other is that the wheels run above the terrain. The same error value in both conditions may lead to different visual effects. In general, the condition when the wheel is lower than the terrain is easier to be noticed than the other condition. So, the condition should be more avoided. \(J_1\) can be modified as equation (6). If \(d_i^+ < d_i^-\), then the error will appear more in the former condition. If \(d_i^+ = d_i^-\), equation (6) can be equated with equation (3).

\[
J_i = \overline{D} \Sigma_i \overline{D}^T
\]

\[
\Sigma_i = diag(\sigma_1, \sigma_2, \ldots, \sigma_6)
\]

\[
\sigma_i = d_i^+ sgn(D_i) + d_i^- sgn(-D_i)
\]

where \(sgn(x)\) is a sign function which can be defined as equation (7).

\[
sgn(v) = \begin{cases} 1, v > 0 \\ 0, v \leq 0 \end{cases}
\]

IV. Experiments

According to the type of 3D terrain, experiences contain two parts: synthetic terrain simulation and real terrain simulation. In simulation system accomplishment, the main software tools refer to VC++6.0 and OPENGL 1.1 and the hardware is the HP workstation XW 8200.

A. Simulation on synthetic terrain

The synthetic scene is sine cross terrain as Fig. 6. The distance of phase difference is 100mm, nearly 10% of the length of rover which is about 1200mm. If there is no phase difference, the terrain can be treated as a 2D surface. For the reason, the scene is designed as the sine cross terrain deliberately.

The simulation interface in synthetic terrain is exhibited in Fig. 7.

With ordinary kinematics model simulation without optimization, the error data is shown in Fig. 8(a). The error will gradually accumulate as time. With the proposed method, the interaction error between the wheels and the terrain is shown in Fig. 8(b). The proposed method eliminates error and error accumulation trend and ensures that the wheels and the terrain can contact tightly all the time.

In the optimization, we adopt simplex search method, because its consuming-time is less than some similar methods such as Newton method, when the variable number is less 10.
and the initial value is closed to the minimum. The initial value comes from the result calculated from kinematics model.

Fig. 6. Sine cross terrain.
Two similar sine terrains lie abreast with different phase. Two white lines are the planning-path for the rover.

Fig. 7. Simulation on synthetic terrain

B. Simulation on the real terrain

In the real terrain simulation, the 3D real terrain (as Fig. 9) is densely reconstructed through two images on the artificial scene like the lunar surface. The virtual reality effect without optimization is shown in Fig. 11. The left three wheels sticks in the terrain surface and the right three wheels fly above the terrain surface.

Fig. 8. Error between wheels and the synthetic terrain
(a) records the error purely based on model kinematics model. (b) records the error with the proposed method. In (a) and (b), the horizontal axis denotes the distance in the forward direction.

With the proposed method, the simulation interface in real terrain is exhibited in Fig. 10 and the error between the wheels and the terrain is shown in Fig. 12. Lunar rover can move attaching to the terrain surface. In the simulation, several main parameters in the simulation are recorded in Fig. 13. The optimization numerical method still adopts simplex search method.
V. CONCLUSION

The paper introduces a navigation system for lunar rover based on virtual reality technology. This system succeeds to simulate the motion of lunar rover according to the planning-path predefined and kinematics model. Because the lunar surface cannot be treated as the 2D surface, the kinematics modeling needs to consider the terrain factor and the kinematics model can be used directly in the simulation. This paper adopts the method in [3,4] to model the relationship between the 3D terrain and rover pose. The optimization method eliminates the error caused by the kinematics model and discretization and improves virtual reality effect.

REFERENCES


