

# 小型月面探査ローバ：Micro-5 及び遠隔制御アルゴリズムに関する研究

研究代表者 研究員 國井 康晴（中央大学理工学部電気電子情報通信工学科）  
 共同研究者 準研究員 多田 興平（中央大学大学院理工学研究科博士後期課程）  
 共同研究者 客員研究員 黒田 洋司（明治大学理工学部機械工学科）  
 共同研究者 客員研究員 久保田 孝（文部科学省宇宙科学研究所）

## 1 はじめに

現在，無人移動探査機を用いた様々な科学観測ミッションが月や火星を対象として検討され，実行されている [1][2][3]。特に，惑星表面上を直接探査する惑星探査ローバは，将来の表面探査ミッションにおいて中心的な役割を担い，多くの地質学者や工学者によって注目されている。1997 年 7 月の Pathfinder 計画，探査ローバ：Sojourner の成功は，記憶に新しい。

一方，我国においても，近年，月に関する科学探査ミッションが複数提案され，実際に，LUNAR-A 計画，SELENE 計画などが検討されている。SELENE 計画では，検討されていた周回衛星とランダ（着陸機）を用いたミッションから着陸探査が分割され，着陸技術試験と表面科学探査をミッション目的とした SELENE-B 計画として検討が始まっている。SELENE-B では，月面の地質探査が行われ，詳細な観測データが取得されることが期待されており，月の起源や進化に関する知見が得られる。惑星上の地質調査を行う際にランダのみを用いた場合，探査範囲は，ランダ周辺のみに限定され，月面上の特定の点のみの観測に留まる。一方，広い地域からの観測データを収集するためには，表面移動機構を備えたローバの使用が要求される。ローバによる観測によって，観測範囲を点から平面へと拡張することが可能になり，より普遍的なデータの取得が可能となる。観測範囲を拡大するために，長い距離を走行することがローバに要求される。しかし月面は，全くの自然地形，すなわち，不整地である。よって，大小の岩や窪み，亀裂等といった様々な地形を超える高い走破性が要求される [1]。

さらに，ローバへの作業指令方式も重要な要素となる。地球から搭載カメラやセンサデータに基づきローバを直接操作した場合，操作者のコマンド生成時間と通信遅れ時間によってローバは，多くの待機時間を有することになり，ミッション時間における実行時間の比率が極端に低下する。このため，可能な限り，ローバの自律機能で対処することが好ましいが，複雑な自然環境においては信頼性に欠くため，不測の事態への対処，観測における細かい作業に対応するために操作者による直接操縦（遠隔操作）が必要不可欠となる。

本稿では，通信時間遅れの存在に起因するローバの待機時間を可能な限り減らし，作業効率の向上を目指す。操縦者によって，ローバが過去に計測した環境データに基づいて生成された経路指令を，ローバが取得した最新データを用いて補正することで経路指令の信頼性を向上させ，時間的な差異を埋めることを検討する。これにより，1 コマンドでの作業量を拡大することでローバの待機時間の減少させることが可能になり，さらにローバ走行中にデータを補正，追加することでローバの待機時間が存在しない遠隔操作手法が実現可能になる。

## 2 Time-Continuous Tele-Driving System

### 2.1 システム概要

通信時間遅れが存在する環境下で操作者がローバを停止させることなく連続操作可能とするためには，ローバが計測または動作した結果のセンサ値が地上の操作者に通信時間分遅れてフィードバックされるのを待つのではなく，

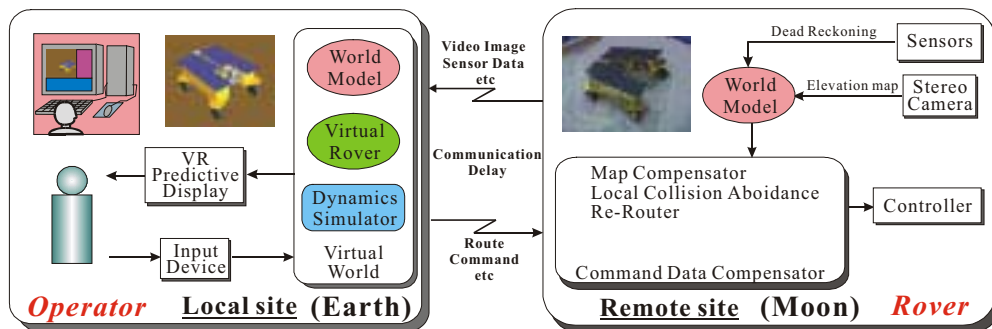


Fig.1 Architecture of Time-Continuous Tele-Driving System

オープンループで操作する必要がある。本システムでは、時間遅れを含めた1つの系でローバと操作者間の制御ループを構成せず、地上側とローバ側で独立した制御ループを構成する。本システムの概略図を Fig.1 に示す。以下、月面上のローバを Remote Rover, 地上の Virtual World 内のローバを Local Rover と呼ぶ。

このシステムにおいて操作者は Remote Rover を直接操作せず、Remote Rover のステレオカメラによって得られた距離画像データとテクスチャから構築される Virtual World 内の Local Rover を操作する。地上の予測表示ディスプレイ (Predictive Display) は未来に到達するであろう地形情報を表示し、操作者はこの画面を見て操作することになる。Local Rover は操作者のコマンドに対してリアルタイムに動作し、コマンドは同時に Remote Rover にも送信される。Remote Rover は通信時間分遅れてコマンドを受け取り、それに従って動作する。本システムでは、Remote Rover と Local Rover は動作に通信時間遅れ分のずれが生じるが、常に一定のずれを維持しながら同期して動作する。

## 2.2 システムの動作

本システムを用いた連続走行の状況を詳細に述べる。Fig.2 にシステムの流れを示す。

- 1 Remote Rover が遠方地形を計測し、地上に向け送信、Remote Rover 内部に World Model を構築する。
- 2 地上側では送られてきたセンサデータに基づき Remote Rover 側と同様に World Model を構築する。操作者は World model を見て走行軌道を生成し、Remote Rover に送信する。Local Rover は走行を開始する。

- 3 Remote Rover は地上から通信時間分遅れて動作する。移動しながら前方地形情報を改新していく。
- 4 操作者は Real Rover から送られた  $T_d[s]$  前の新たな環境データに基づき構築された Virtual World 中で走行軌道を決定する。
- 5 Remote Rover は走行軌道データを受け取った時点で最新の環境情報を取得する。そして、走行軌道を決定した際に用いた以前送信した環境情報と最新の環境情報と比較し、ずれがある場合 Remote Rover 自身が走行軌道を修正する。

上記の 3 ~ 5 を繰り返すことにより、Remote Rover は周囲環境が自己が認識する環境と差異が生じた場合にも休止することなく連続的に移動することが可能となる。

## 3 軌道補正アルゴリズム

前節で述べたように、本システムによってローバが月面上で連続走行する際に生じる状況として、操作者が操作を行う Virtual World 中に構築する World Model と月面上の Remote Rover 自身が内部に保持している最新の World Model 間に、通信時間遅れに起因する時間的・空間的ずれが発生することが考えられる [5]。

### 1 通信時間遅れによるずれ

操作者が Local Rover にコマンドを下す時刻を  $T[s]$  とすると、 $T[s]$  時の地上の World Model は Remote Rover が  $T - T_d[s]$  時に獲得したデータにより構築されたものであり、 $T[s]$  時に生成したコマンドが Remote Rover に到着するのは  $T + T_d[s]$  時である。つまり、ローバに届く走行軌道データは、操作者が  $2T_d[s]$  前に

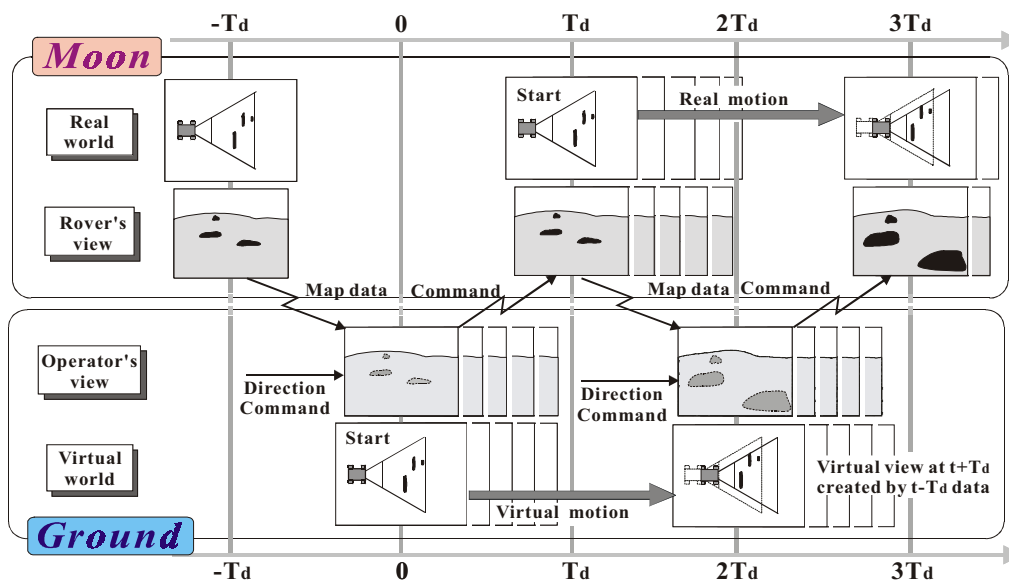


Fig.2 Time chart of Time-Continuous Tele-Driving System

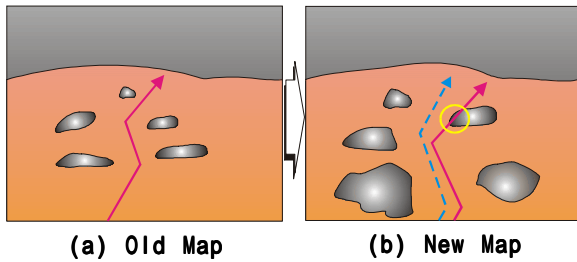


Fig.3 Distortion Compensation

獲得された環境データにより構築された World Model 中で行われたものである。この 2Td[s] の時間差間に Remote Rover はかつて遠方に存在した領域に向けて常に直近の環境画像を取得しながら移動し、World Model を更新する。Remote Rover は前進することにより障害物までの計測距離が小さくなるため、Real Rover 内部に保持し更新される World Model は移動するごとに精度の良いデータにより再構築される。

## 2 自己位置誤差によるずれ

着陸機を離れて行動する場合、ローバは自己位置をデッドレコニングにより取得すると考えると、操作者がコマンドを生成するのに用いた旧マップと Remote Rover がコマンドを受け取った時点で獲得した最新のマップ間で、Remote Rover 自身が不整地を移動することによる誤差が加わり、障害物までの相対距離のずれはさらに拡大すると考えられる。

これら 2 つを踏まえた結果、Real Rover が障害物へ接近すると、新旧の周囲情報データに差異が生じ、対処しなければ衝突してしまう状況に陥ることが予想される (Fig.3)。

そこで本システムでは、Fig.3(b) 内の点線矢印のように操作者が過去に生成した走行軌道データを Remote Rover の現在の環境に適応させる方法を用いる。

まず、Remote Rover が移動中随時更新する最新の環境データとコマンド生成に用いた古い環境データ間においてずれ・歪みを補正する「歪み補正変換行列」を求める。その後、その変換行列を過去の走行軌道データに乗ずることによりローバ自身が補正走行軌道データを生成する。

### 3.1 軌道補正フロー

周囲環境情報の違いを計算によって求め、その結果を走行経路補正に用いる Remote Rover 側に搭載するシステムの流れを Fig.4 に示す。

#### Step1 マップの構築・更新

Remote Rover はステレオカメラから濃淡画像を取得し、距離画像データを出力する。得られた距離画像データより DEM を構築する。Remote Rover は移

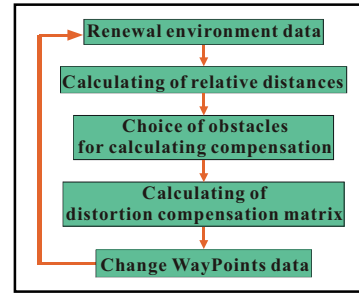


Fig.4 Flow of Continuous Command Data Compensation

動しながら絶対座標系でマップを重ね、広げていく。

#### Step2 相対距離の計算

相対距離の計算構築した World Model 中の Remote Rover の位置座標と Remote Rover のカメラ視野に入った障害物の位置座標から、Remote Rover から各障害物までの相対距離を求める。

#### Step3 対象障害物の選択

障害物との相対距離から Remote Rover と障害物との位置関係によって補正変換エリアを決定し、歪み補正変換行列算出に用いる障害物を選択する。

#### Step4 歪み補正変換行列の算出

Step3 で選択された新しい World Model 内の障害物特徴点データを抽出する。そして、ウェイポイントを決定した時に構築した古い World Model 中から対応する障害物データを検索し、位置データを抽出する。この 2 組のデータに最小二乗法を用い、歪み補正変換行列を求める。

#### Step5 ルート補正

Remote Rover 自身が歪み補正変換行列をあらかじめ指定されたウェイポイント列に乗ずることにより、走行経路を変更する。

### 3.2 World Model 歪み補正

Step4 の歪み補正変換行列の導出方法を述べる。

RemoteRover に搭載したステレオカメラのデータにより構築される Digital Elevation Map (DEM) は三次元データであるが、ここでは歪み補正変換行列計算は二次元平面データにより求めるものとし、線形であると仮定する。走行軌道データ生成時の World Model 内の特徴点座標の集合を  $X(x, y)$  とし、Real Rover が走行軌道データを受信したときの最新画像で得られる作業空間内の対応点座標の集合を  $\bar{X}(x', y')$  とする。この時  $X$  と  $\bar{X}$  は、歪み補正行列  $A$  を用いて

$$\bar{X} = XA \quad (1)$$

$\bar{X}$ : コマンド受取時の最新の対応点座標

$X$ : 操作者が送信した時の古い特徴点座標

$A$ : 歪み補正変換行列

と表せる。ここで歪み補正変換行列は以下のように Householder の鏡像変換を用いて最小二乗解を求めることにより得られる。

$$XP = QR \quad (2)$$

( $P$ : 置換行列,  $Q$ : 直交行列,  $R$ : 上三角行列)

$$A = P(R^{-1}(Q^T \bar{X})) \quad (3)$$

この写像関数を操作者が指示した走行軌道データに乗ずることにより, 修正された走行軌道が得られる。

$$Wp_{new} = Wp_{old}A \quad (4)$$

$Wp_{new}$ : 補正変換されたウェイポイント列

$Wp_{old}$ : 操作者が設定したウェイポイント列

### 3.3 特徴点選択アルゴリズム

Step3 の歪み補正変換行列算出に用いる対象障害物の選択方法を述べる。

Remote Rover はステレオカメラにより環境情報を取得し, 一度捕らえた特徴点をトラッキングしていく能力を持つとする。特徴点データは距離が遠いほどカメラ誤差が大きいため, 近くの障害物を避ける際に遠方の特徴点の誤差を含めた補正行列により補正を行うと, 作成したマップ全体に対する歪み補正行列を求めることとなるため, 最も回避しなければならない最近傍の障害物に対する補正は正確な変換とはならない。よって, 近傍に障害物が存在する状況では, 補正変換行列を求めるときに用いる特徴点は誤差の少ない, Remote Rover から最近傍のデータのみを用いて計算する。特徴点を選択することで Remote Rover の直近の障害物だけのマップに対して重点をおいて補正を行い, より確実に操作者の意図に近い軌道に変更する。Fig.5 に障害物の選択エリア図を示し, 以下に選択条件を述べる。

- 1 Area1 に障害物が入ったときは, その障害物を選択する。また, Area1 に障害物がない場合, Area2 に存在する障害物を選択する。
- 2 過去に Remote Rover の視野エリア中の Area1 内に入った障害物は直近を通過したため測定距離誤差が小さく, 正確な位置情報を獲得しており, 位置座標の信頼性が高いため過去に通過したランドマークとして選択する。しかし, 常に走行誤差が加わり, データの信頼度が下がるため, 距離が大きくなると選択をやめる。

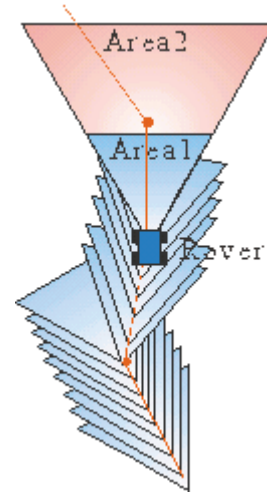


Fig.5 Search area for guide to calculate distortion compensation



(a) Image sent from rover to ground

(b) Image captured at Td[s] later from image(a)

Fig.6 Difference between two camera images in turn around communications

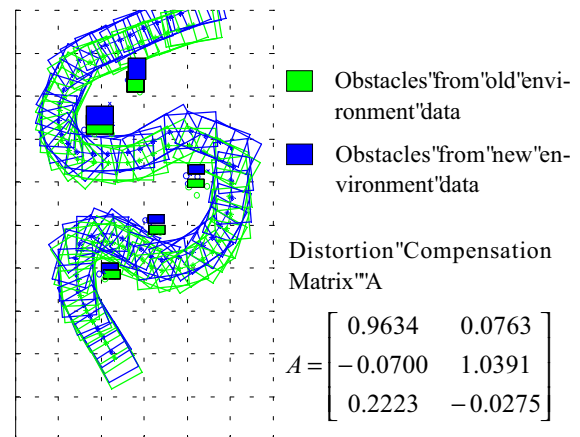


Fig.7 An example of calculating command Data compensation using actual stereo camera data

### 4 軌道補正実験

ここでは, 以下の実験について述べる。

- 1) ステレオカメラによる 2 地点の実計測データによる軌道補正シミュレーション
- 2) あらかじめウェイポイントデータを与えた Remote Rover の連続的な走行軌道補正機能の検証



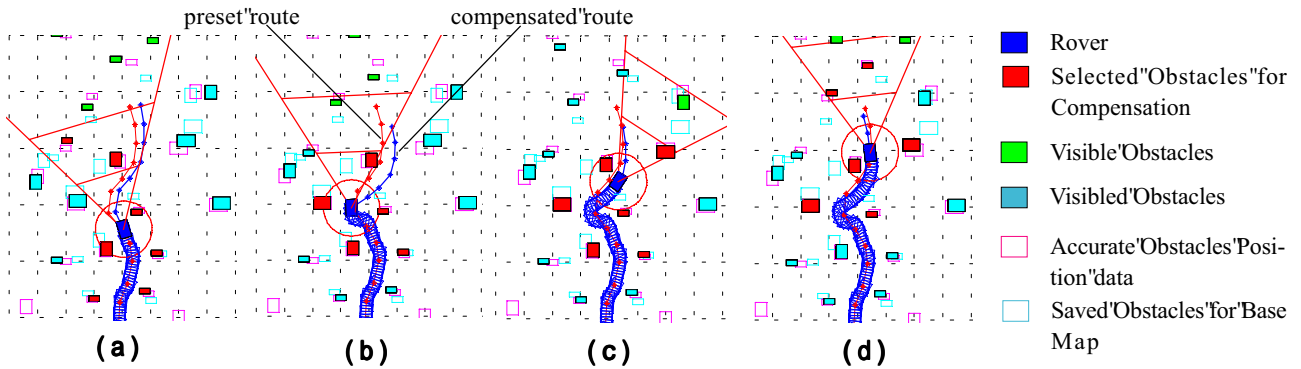


Fig.8 Simulation Result of Path Tracking with Command Data Compensation

### 3)実システムを想定した 3DCG シミュレータによる検証と実機実験

#### 4.1 実測データによる軌道補正計算

Fig.6 にシミュレーションに用いた画像を示す。Fig. 6(b) は (a) より 1m 近づいた時の画像である。Fig.6(a) の特徴点と Fig.6(b) の対応点の距離データによりそれぞれの WorldModel を構築する。そして、歪み補正変換行列を求め、走行軌道を補正する。Fig.7 に補正結果を示す。

移動前と移動後の距離画像データから構築した World Model をワールド座標上で重ねると、障害物の位置にはずれが生じた。障害物の位置のずれ方から 2 つの World Model 間には歪みが生じている。その結果、オペレータが古い World Model を用いて設定した走行軌道のままでは、Remote Rover は障害物に衝突してしまうことになる。

歪み補正変換行列を求め、走行軌道補正を行った結果、障害物の歪み・ずれに対応して走行軌道が補正変換され、障害物に衝突しない軌道に変更された。

#### 4.2 連続走行軌道補正シミュレーション

##### 4.2.1 計算条件

シミュレーションの各種パラメータを以下に述べる。

- ・シミュレーション空間：前方 8m× 左右 2m
- ・カメラ視野角：60°
- ・カメラ誤差：1m につきランダムに 10%  
(実際のカメラ誤差は一定方向に 1.7%)
- ・対象障害物選択範囲
- ・カメラ視野内
  - Area1 カメラ前方 1.0m 以内
  - Area2 カメラ前方..0m 以内
- ・過去の走行軌道周辺
  - Real Rover から 1.2m 以内
- ・補正頻度
  - 10cm 移動するごとに 1 回行う
- ・最大旋回角度

Area1 に障害物が入っている時 20°

Area2 に障害物が入っている時 10°

##### 4.2.2 シミュレーション結果

Fig.8 に連続軌道補正シミュレーションの結果を示す。この結果は、スタート後ある程度時間が経過した状況を示しており、ウェイポイントを設定する際、障害物の位置誤差が大きかったエリアを通過している様子である。

円で囲まれた長方形がローバであり、その前方にはカメラの視野エリアを示し、特徴点抽出エリアを横線で示してある。薄色の線は操作者により設定された走行軌道である。濃い色の線はローバが歪み補正変換行列を求めて新しく設定された軌道である。障害物は条件により色が変化し、色の濃い順に歪み補正変換行列を計算するのに用いた障害物、カメラ視野に入っている障害物、過去にカメラ視野に入ったが位置データの信頼性が低い障害物となっている。障害物の位置はカメラからの距離に応じた誤差を与えており、ローバから遠方の障害物ほど 1 ステップ毎に大きく位置が動いている。

Fig.8(a) では、操作者の設定した走行軌道では、正面に位置する障害物に衝突してしまう経路となっている。Fig.8(a) の状況では、Area1 に障害物がないため Area2 から障害物を選択しており、走行軌道は緩やかに補正されている。歪み補正変換計算は過去に Area1 内に入った障害物の位置データも 5 つ使用し、安定した補正を行っているためである。

Fig.8(b) では、Real Rover の Area1 の右前方に障害物が入ったため、歪み補正変換計算は Area1 内の障害物データを用いて行われている。その結果、よりローカルな補正計算が行われ、Fig.8(b) では Fig.8(a) よりも大きく右方向に経路を変更している。

Fig.8(d) を見ると、最終的には操作者により設定された走行軌道よりも大きく障害物を回避したことがわかる。

以上の結果より、本システムにより Remote Rover は

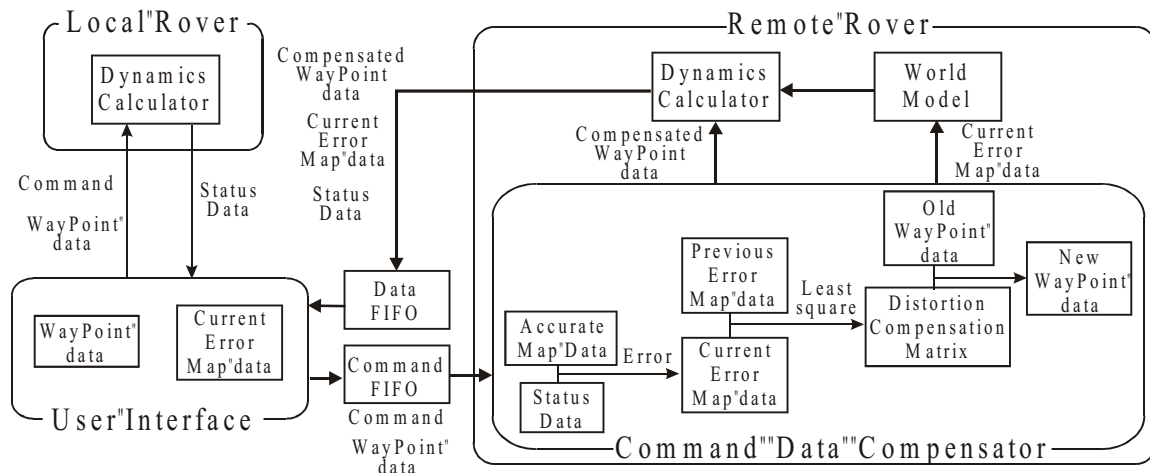


Fig.9 Architecture of Tele-Driving Simulator

直面する環境データに変化が生じて、その状況に応じて操作者の指示した走行経路を変更し、衝突しそうになった障害物を避けることができた。特に、ローバの近傍に障害物が入った場合は、障害物の直前で大きく進路を変更し、確実に衝突を避けることができた。

### 4.3 実システムを想定した実験

走行軌道補正アルゴリズムを用いて連続走行する Remote Rover と相互にコマンド、データを通信し、走行速度、カメラの視野角、通信時間遅れ幅などパラメータによる影響を検証・評価するために、3次元シミュレータを構築した。

#### 4.3.1 連続走行軌道補正 Tele-Driving Simulator

シミュレーションシステムの内部構成を Fig.9 に示す。本システムは User Interface, Local Rover, Remote Rover

の3つのプロセスに分かれ、それぞれが Socket 通信によりデータの送受信を行っており、プロセスごとに別のマシン上で実行することが可能である。シミュレータのユーザインタフェースをそのまま実ローバのコントローラとして使用できるように通信プロトコルを設定した。また、時間遅れを人工的に生成するために、FIFO バッファを2つ用いている。

3次元グラフィックライブラリとして Open Inventor を利用した連続走行軌道補正シミュレータの GUI 画面を Fig.10 に示す。シミュレーション時に Local 側, Remote 側として2台のグラフィックワークステーション上で実行すると、実際の月・地球間環境を模した遠隔操作環境とすることができる。Remote Rover, Local Rover それぞれのプロセスでは、カメラ視点画像と鳥瞰視点ウイ

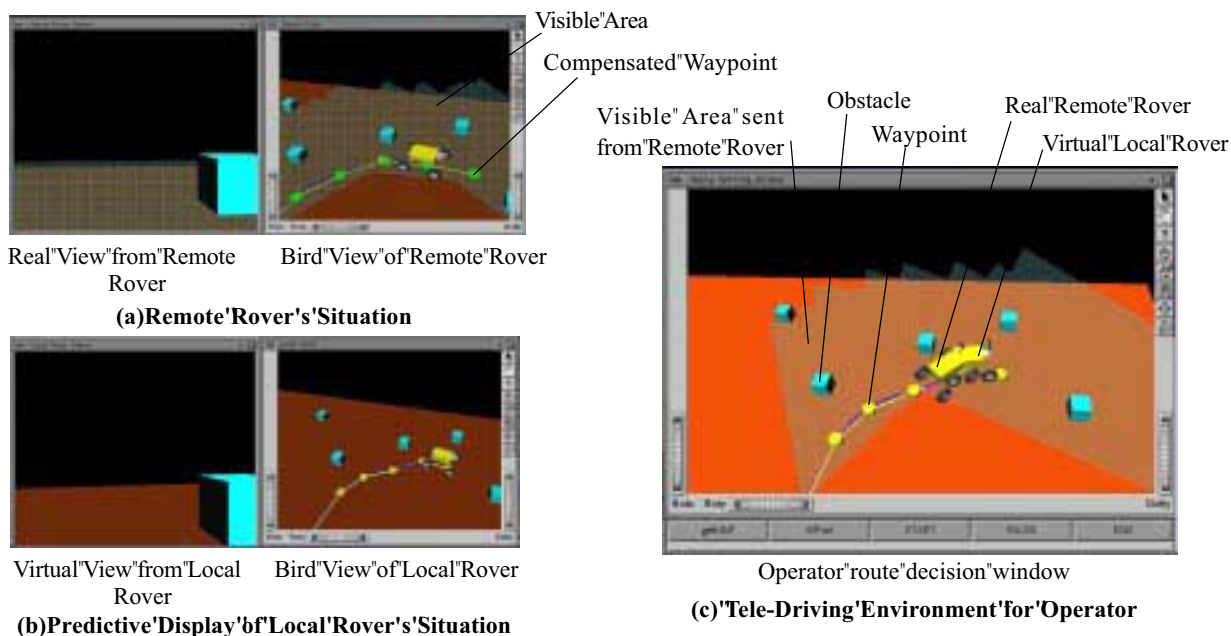


Fig.10 Graphical User Interface of Tele-Driving Simulator

ンドウを表示する。User Interfaceにはウェイポイント設定ウィンドウが表示される。シミュレータ上では、特徴点データにより表示する障害物を50cm四方の立方体で表現している。ウェイポイント指定ウィンドウ中には、Remote Roverのカメラにより獲得した環境を随時表示し、移動するとともに取得エリアが拡大する様子と理想的な動作計算結果のLocalRoverと月面上のRemote Roverから送信された位置・姿勢データにより再構成されたRemote Roverが表示される。この2台の位置には往復の時間遅れ分のずれが存在し、Local RoverとRemote Roverは $2Td[s]$ の時間差で表示される。Remote Roverの鳥瞰視点ウィンドウでは、操作者から指定されたウェイポイントと補正変換により変換されたウェイポイントの2種類を表示する。操作者は予測表示であるLocalRoverのカメラ視点画像とウェイポイント指定ウィンドウによりローバの周囲環境状況を判断し、操作を行う。

#### 4.3.2 Tele-Driving Simulator 動作結果

テスト条件として、ローバ搭載カメラ視野を $60^\circ$ 、速度を10cm/s、片道通信時間遅れを5秒とした。

Remote Roverが獲得した環境情報に基づいて再構築された仮想環境中において、ウェイポイントをRemote Roverが獲得した環境外に指定すると、補正用基準マップを構築した時点では地形状況を把握していない環境に走行していくため、走行軌道補正機能を用いることができない。よって、走行軌道補正アルゴリズムを用いてRemote Roverを広範囲に走行させるためには、前方地形状況取得時のカメラ視野角を大きくする、もしくはスキャンすることにより、広いエリアのマップデータを取得しておけば良いことがわかる。

#### 4.3.3 実機を用いた走行実験

実環境に対する適応能力を示すためにテストベッドローバ「Dekabot」を作成し、本手法の適用実験を行うことを計画している。Fig.11に「Dekabot」の外観を示す。走行機構として4WS, 4WDを採用し、視覚センサとして3眼ステレオビジョンシステム「Triclops」を搭載している。



Fig.11 Testbed Rover “Dekabot”

OSとして、RT-Linuxを採用し、TCP/IPと無線LANを用いてsocket通信によりデータ送受信を行う。操作環境は前節で述べたTele-Driving SimulatorのGUIをコントローラに改良し、シミュレータと同じ操作環境で実験を行う。

## 5 おわりに

本論文では、月面探査ローバにおける長距離無停止連続走行を目的とした「Time-Continuous Tele-Driving System」を提案した。そして、本システムでの基本機能である操作者とローバ間において通信時間遅れに起因する環境マップのずれ・歪みを補正する手法を述べた。また、連続走行軌道補正シミュレーションにより、ローバに搭載する歪み補正変換アルゴリズムが直面する障害物との距離に応じた局地的な変換を行なうことにより走行軌道を最適に補正し、有効に機能することが確認された。また、通信時間遅れ等様々な条件を見立て、実際の地上と月面の環境を想定し、実機との接続を考慮したTime Continuous Tele-Driving Simulatorが走行軌道補正アルゴリズムを搭載した状態で非常に有効に動作する事が確認された。

走行軌道補正を行いながら広範囲をより高い自由度で走行するには、ローバ搭載カメラをスキャンさせる等、広範囲に環境情報を獲得する手法について検討を進めていく予定である。

## 参考文献

- [1] J. L. Loch, D. Desai: “Moose on the Loose: Toward Extended Mission Autonomy for Robotic Exploration of Planetary Surface,” Proc. of ICAR '93, pp.97-101, 1993.
- [2] G. Giralt: “Remote Intervention Robot Autonomy and Real World Application Cases,” Proc. of IEEE Int. Conf. on R&A, pp.541-547, 1993.
- [3] R. Chatila, R. Alami, et al.: “Planet Exploration by Robots: From Mission Planning to Autonomous Navigation,” Proc. of ICAR'93, pp.91-96, 1993.
- [4] L. Matthies et al.: “Mars Micro Rover Navigation: Performance Evaluation and Enhancement,” Proceedings of the IEEE/RSJ International Conference on Robots and Systems (IROS), Pittsburgh PA, 1995.
- [5] Y. Kuroda, K. Kondo, K. Nakamura, Y. Kunii, T. Kubota: “Low Power Mobility System for Micro Planetary Rover “Micro5”,” Proc. of 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, pp.77-82, 1999.